ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εφαρμογή Μηχανικής Μάθησης στην Ανάλυση Άποψης Κειμένων στον Θεματικό Τομέα των Τουριστικών Επιχειρήσεων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των

Κωνσταντίνου Ο. Γκιόκα **Οδυσσέα Ο. Γκιόκα**

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2015

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

Εφαρμογή Μηχανικής Μάθησης στην Ανάλυση Άποψης Κειμένων στον Θεματικό Τομέα των Τουριστικών Επιχειρήσεων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

των

Κωνσταντίνου Ο. Γκιόκα **Οδυσσέα Ο. Γκιόκα**

Επιβλέπων: Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 6η Μαρτίου 2015.

.....	.....	.....
Στέφανος Κόλλιας	Ανδρέας-Γεώργιος Σταφυλοπάτης	Γιώργος Στάμου
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2015.

.....
Οδυσσέας Ο. Γκιόκας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

.....
Κωνσταντίνος Ο. Γκιόκας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η παρούσα διπλωματική πραγματεύεται το πρόβλημα του Sentiment Analysis δηλαδή την αυτόματη κατηγοριοποίηση ενός κειμένου ως θετικό ή αρνητικό με γνώμονα την άποψη του συγγραφέα πάνω στο θέμα του κειμένου και ιδιαίτερα σε κείμενα που αφορούν κριτικές ξενοδοχείων. Αφού οριστεί το θεωρητικό υπόβαθρο του προβλήματος, επιλέγονται τρεις μέθοδοι (αλγόριθμοι) για να εκπαιδευτεί ένα σύστημα σε αυτήν την αυτόματη κατηγοριοποίηση. Συγκεκριμένα επιλέγονται ο αλγόριθμος Naive Bayes, μία τροποποίηση του Hidden Markov Model που ονομάζεται Lexicalised Hidden Markov Model Integrating Part-of-Speech και Νευρωνικά Δίκτυα. Στον αλγόριθμο Naive Bayes δοκιμάσαμε μερικές παραλλαγές του, με διαφοροποιήσεις κάθε φορά στο ποιες λέξεις συμπεριλαμβάνονται αναφορικά με την συχνότητα τους και το μήκος τους, αν χρησιμοποιούνταν σκέτες λέξεις ή και n-grams και το πως γινόταν ο χωρισμός των λέξεων μεταξύ τους. Στην τροποποίηση του Hidden Markov Model επιλέχθηκε ένα σύνολο από χαρακτηριστικά (tags) και εφαρμόστηκε ένα πιο ειδικό πεδίο του Sentiment Analysis, το Aspect Based Sentiment Analysis. Στα Νευρωνικά Δίκτυα εφαρμόστηκαν υλοποιήσεις για one-layer και three-layer perceptrons και έγιναν πειράματα με διαφορετικές τιμές στις παραμέτρους του μοντέλου ώστε να επιτευχθούν τα καλύτερα αποτελέσματα.

Για να δοκιμαστούν αυτοί οι αλγόριθμοι χρησιμοποιήθηκαν κυρίως κριτικές ξενοδοχείων οι οποίες αντλήθηκαν από το booking.com και παρατίθενται για αναφορά, αλλά και το ευρέως χρησιμοποιούμενο στο Sentiment Analysis σύνολο δεδομένων από κριτικές ταινιών του imdb.com των Pang και Lee.

Ως συμπέρασμα καταλήξαμε ότι για να επιτευχθεί μία πολύ ικανοποιητική απόδοση και ακρίβεια από το μοντέλο σε κριτικές ξενοδοχείων είναι αρκετός ένας απλός αλγόριθμος όπως ο Naive Bayes με την ακρίβεια προβλέψεων να φτάνει μέχρι και 94.5%. Αν απαιτείται ανάλυση υποχαρακτηριστικών του κειμένου τότε μπορεί να χρησιμοποιηθεί το Hidden Markov Model αλλά με χαμηλότερη ακρίβεια προβλέψεων. Τα Νευρωνικά Δίκτυα δείχνουν να μην ξεπερνούν σε ακρίβεια τον αλγόριθμο Naive Bayes παρά τη δυσκολία χρήσης τους.

Λέξεις Κλειδιά: Μηχανική Μάθηση, Ανάλυση Άποψης, Κατηγοριοποίηση Βασισμένη στην Άποψη, Ανάλυση Άποψης Βασισμένη σε Χαρακτηριστικά, Naive Bayes, Κρυφό Μοντέλο Markov

Abstract

In this dissertation we consider the problem of Sentiment Analysis, which refers to automatically classifying documents as positive or negative with regards to the writer's opinion on the central subject of the document and especially we consider the application of the problem in hotel reviews. After the theoretical background is specified, three distinct methods (algorithms) are chosen to train a system to perform such an automatic classification. Specifically, the chosen algorithms are Naive Bayes, a variation of a Hidden Markov Model called Lexicalised Hidden Markov Model Integrating Part-of-Speech and Neural Networks. Considering Naive Bayes, we tried different versions of the algorithm, differentiating by which words are allowed with regards to their frequency and length, by whether single words were used or n-grams were included and the way the words were actually split. In the variation of Hidden Markov Model, we chose a set of features (tags) and we considered the more specific field of Sentiment Analysis called Aspect Based Sentiment Analysis. In Neural Networks we structured both one layer and three layer perceptrons and experiments were conducted whilst tweaking the parameters of the system to achieve the best possible results.

In order to test those algorithms we used mainly hotel reviews that were scraped of booking.com website and are available for reference, but additionally we used the dataset that is the most popular in Sentiment Analysis, the dataset of movie reviews from imdb.com website by Pang and Lee.

In the end, we concluded that in order to achieve a good performance and precision, a simple algorithm like Naive Bayes is sufficient with precision percentages reaching the number of 94.5%. If an aspect based analysis on the text is required then a Hidden Markov Model is advised though the precision will be lower. Neural Networks seem to not exceed Naive Bayes' performance, even though they are harder to use.

Keywords: Machine Learning, Sentiment Analysis, Sentiment Detection, Aspect Based Sentiment Analysis, Naive Bayes, Hidden Markov Model

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός και αντικείμενο της διπλωματικής εργασίας

Είναι πλέον γνωστό ότι τα δεδομένα που υπάρχουν στο διαδίκτυο αυξάνονται καθημερινά και έχουν διαμορφώσει έναν τερατώδη όγκο από πληροφορία ο οποίος είναι αδύνατο να επεξεργαστεί μόνο από τον άνθρωπο. Συνεπώς ο τομέας του data mining έχει γίνει εξαιρετικά δημοφιλής τα τελευταία χρόνια και δημιουργούνται πολλαπλές εφαρμογές πάνω σε αυτόν. Μία από αυτές είναι να μπορείς να κατηγοριοποιήσεις κείμενα με βάση κάποια γνωρίσματα και ονομάζεται Αυτόματη Κατηγοριοποίηση Κειμένου ΑΚΚ (Automated Text Categorization).

Σε αυτή τη διπλωματική εργασία εξετάζουμε αλγορίθμους και μεθόδους κατηγοριοποίησης με βάση την άποψη (θετική/αρνητική) για χρήση σε κείμενα που αφορούν τουριστικές επιχειρήσεις. Τα κείμενα που έχουν συλλεχθεί για επαλήθευση των μεθόδων είναι κυρίως κριτικές ξενοδοχείων και είναι γραμμένα είτε στην ελληνική, είτε στην αγγλική γλώσσα.

Αντιλαμβάνεται κανείς πως η εφαρμογή σε αυτό το πεδίο είναι ιδιαίτερα ενδιαφέρουσα για την Ελλάδα, στην οποία ο τουρισμός είναι από τις κυριότερες πηγές εσόδων της. Έτσι, υπάρχει πολύ μεγάλος όγκος από κριτικές για τουριστικές επιχειρήσεις τόσο σε ελληνικά, όσο και σε διεθνή domains.

Η κατηγοριοποίηση κειμένων (tweets, σχολίων σε blogs, άρθρων) με βάση την άποψη μπορεί να βοηθήσει την επιχείρηση να εστιάσει στις αρνητικές κριτικές και να εντοπίσει τα μειονεκτήματα της, είτε να εστιάσει στις θετικές ώστε να αντιληφθεί τι λειτουργεί σωστά και να φροντίσει να το εξασφαλίσει. Σε γενικότερο επίπεδο, μπορεί να μας δώσει μία στατιστική η οποία θα δείξει τα καλά και τα κακά των τουριστικών επιχειρήσεων μίας χώρας (π.χ. Ελλάδα) για να δούμε τι μπορεί να αλλάξει από θεσμούς / νομοθεσίες / ελέγχους και να διασφαλίσουμε μεγαλύτερη ποιότητα τουριστικών υπηρεσιών.

Για να επιτευχθεί αυτή η Κατηγοριοποίηση με Βάση την Άποψη (ΚΒΑ)

στα κείμενα των τουριστικών επιχειρήσεων, δοκιμάσαμε διάφορες μεθόδους. Αναφορικά οι μέθοδοι που χρησιμοποιήθηκαν ήταν: ο αλγόριθμος Naive Bayes, με χρήση HMM, με χρήση νευρωνικού δικτύου. Δοκιμάστηκαν διάφορες παραλλαγές / τροποποιήσεις και ρύθμιση παραμέτρων στις παραπάνω μεθόδους και περιγράφονται αναλυτικά στη συνέχεια.

Επίσης, κατά τη διαδικασία έρευνας και υλοποίησης των μεθόδων αυτών καταφέραμε να εντοπίσουμε διάφορα εργαλεία που βοηθάνε στην κατηγοριοποίηση και επεκτείναμε τη χρησιμότητα της μεθόδου για να επιτύχουμε Aspect-based Sentiment Analysis δηλαδή την ΚΒΑ συγκεκριμένων γνωρισμάτων σε ένα κείμενο και θα αναλυθεί στη συνέχεια.

1.2 Οργάνωση της διπλωματικής εργασίας

Η παρούσα εργασία είναι οργανωμένη με την εξής δομή:

Το κεφάλαιο 1 (παρόν) αφορά το αντικείμενο και την οργάνωση της διπλωματικής.

Στο κεφάλαιο 2 αναφέρουμε ό,τι έχει να κάνει με το θεωρητικό υπόβαθρο που απαιτήθηκε για την περαίωση της διπλωματικής και το οποίο είναι χρήσιμο για την κατανόηση αυτής από τον αναγνώστη. Πιο συγκεκριμένα ορίζονται οι βασικές θεωρητικές έννοιες (μηχανική μάθηση, sentiment analysis) και περιγράφονται αναλυτικά οι αλγόριθμοι που χρησιμοποιήθηκαν (Naive Bayes, Hidden Markov Model, Νευρωνικά Δίκτυα).

Στο κεφάλαιο 3 σκιαγραφούμε τον τρόπο υλοποίησης και όλα τα πρακτικά ζητήματα που προκύπτουν κατά την εφαρμογή των αλγορίθμων, παρουσιάζοντας παράλληλα επεξηγηματικά κομμάτια κώδικα. Τέλος, αναφερόμαστε σε δύο υποεργαλεία που είναι σημαντικά για τις υλοποιήσεις (Stemmer, Part-of-Speech Tagger).

Στο κεφάλαιο 4 παραθέτουμε μία σύντομη περιγραφή των συνόλων δεδομένων που χρησιμοποιήθηκαν καθώς και τα πειραματικά αποτελέσματα που προέκυψαν από την εκτέλεση των αλγορίθμων μας σε αυτά. Γίνεται αναφορά και επεξήγηση στον τρόπο με τον οποία μετράμε την αποτελεσματικότητα και στις διάφορες μετρικές της.

Στο κεφάλαιο 5 εξάγουμε κάποια τελικά συμπεράσματα που αφορούν τους αλγορίθμους μας και τα αποτελέσματα που βγάζουν στα εξεταζόμενα σύνολα δεδομένων.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

2.1 Μηχανική Μάθηση

Η Μηχανική Μάθηση (Machine Learning) είναι ένα πολύ σημαντικό κεφάλαιο της τεχνητής νοημοσύνης και αφορά αλγορίθμους και μεθόδους που χρησιμοποιούνται ώστε οι υπολογιστές να μπορούν να «μαθαίνουν». Πεδία τα οποία χρησιμοποιεί η μηχανική μάθηση είναι η στατιστική, η θεωρία πληροφορίας και η γνωστική επιστήμη. Υπάρχουν τρεις βασικές κατηγορίες μηχανικής μάθησης:

- Η επιτηρούμενη μάθηση (supervised learning). Αυτός ο τύπος μηχανικής μάθησης αφορά μεθόδους και αλγορίθμους στους οποίους δίνουμε ένα σύνολο εισόδων μαζί με επιθυμητές εξόδους ώστε το σύστημα να προσαρμοστεί με βάση αυτά τα δεδομένα. Αυτό είναι το λεγόμενο σύνολο εκπαίδευσης (training set). Έπειτα το σύστημα αποφαίνεται για δεδομένα στα οποία δε γνωρίζουμε την έξοδο τους.

- Η μη επιτηρούμενη μάθηση (unsupervised learning). Σε αυτήν το σύστημα δοκιμάζεται χωρίς να έχει γίνει εκπαίδευση με ταξινομημένα δεδομένα.

- Ενισχυτική μάθηση (reinforcement learning), όπου το σύστημα μαθαίνει αλληλεπιδρώντας με το περιβάλλον του.

Εμείς θα ασχοληθούμε με αλγορίθμους επιτηρούμενης μάθησης. Μηχανική Μάθηση μπορεί να υπάρχει σε πληθώρα επιστημονικών και μη εφαρμογών, όπως πρόβλεψη ιδιοτήτων υλικών ανάλογα με κάποια χαρακτηριστικά της δομής τους, την αναγνώριση ομιλίας ή την Κατηγοριοποίηση με Βάση την Άποψη. Η απόδοση ενός αλγορίθμου μηχανικής μάθησης εξαρτάται από πολλούς παράγοντες, όπως ο αλγόριθμος που θα επιλεχθεί, οι παράμετροι, η ποιότητα των δεδομένων (τόσο training όσο και test) και το πεδίο στο οποίο θα εφαρμοστεί.

2.2 Sentiment Analysis

2.2.1 Document Classification

Το πρόβλημα της Κατηγοριοποίησης Κειμένου (Document Classification) αφορά την απόδοση μίας ή περισσότερων κατηγοριών ή κλάσεων σε ένα δεδομένο κείμενο. Αυτό μπορεί να γίνει είτε χειροκίνητα, είτε αλγοριθμικά. Η Αυτόματη Κατηγοριοποίηση Κειμένου αφορά την αλγοριθμική επίλυση αυτού του προβλήματος.

Χρήσεις της Αυτόματης Κατηγοριοποίησης Κειμένου συναντάμε σε διάφορα πεδία, όπως συστήματα διαχείρισης περιεχομένου (Content Management Systems, CMS), αναζήτηση με βάση τα συμφραζόμενα (contextual search), ανάλυση κριτικών προϊόντων (product review analysis), φιλτράρισμα ανεπιθύμητης αλληλογραφίας (spam filtering) αλλά και στην εξόρυξη άποψης από κείμενο (opinion mining).

2.2.2 Sentiment Analysis

Η Κατηγοριοποίηση με Βάση την Άποψη (ΚΒΑ) είναι γνωστή ως Sentiment Analysis ή Opinion Mining στα αγγλικά και αφορά την κατηγοριοποίηση ενός κειμένου με βάση δύο πολωμένα αντίθετα αποτελέσματα (συνήθως 'θετικό' ή 'αρνητικό'). Είναι άρρηκτα συνδεδεμένη με την επεξεργασία φυσικής γλώσσας (Natural Language Processing, NLP). Δεν πρέπει να μπερδεύεται με την Κατηγοριοποίηση με Βάση το Συναίσθημα (emotion analysis) η οποία αφορά τη συναισθηματική κατάσταση στην οποία βρισκόταν ο συγγραφέας όταν έγραφε το κείμενο, η οποία μπορεί να είναι 'λυπημένος', 'θυμωμένος', 'μπερδεμένος' κ.ο.κ.

Η ΚΒΑ ορίζεται ως προς τη θετική ή αρνητική συμπεριφορά ενός συγγραφέα ή ομιλητή. Η συμπεριφορά αυτή μπορεί να οριστεί με τρεις διαφορετικούς τρόπους:

- Η κρίση του ή τελική του απόφαση για το αντικείμενο στο οποίο αναφέρεται.
- Η κατάσταση στην οποία βρίσκεται ενόσω γράφει το κείμενο.
- Η άποψη που θέλει να περάσει στον αναγνώστη.

Μερικές από τις πρώτες εργασίες πάνω στον χώρο του Sentiment Analysis είναι αυτές του Turney [Tur02] και των Pang, Lee και Vaithyanathan [PLV02] που εφάρμοσαν διαφορετικές μεθόδους αναγνώρισης της πολικότητας (polarity) της απόψεως σε κριτικές προϊόντων και ταινιών αντίστοιχα. Η δουλειά αυτή ήταν σε επίπεδο κειμένου. Κάποιος μπορεί να κατηγοριοποιήσει την πολικότητα της απόψεως σε μία ευρύτερη κλίμακα, όπως προσπάθησε να κάνει ο Pang και ο Snyder μεταξύ άλλων οι οποίοι επέκτειναν την κατηγοριοποίηση

μίας κριτικής ως θετικής ή αρνητικής γνώμης σε πρόβλεψη βαθμολογιών με αστέρια (3, 4 ή παραπάνω) [SB07].

Ενώ σε πολλά στατιστικά μοντέλα, η κλάση της ουδέτερης άποψης (neutral class) αγνοείται, πολλοί ερευνητές ισχυρίζονται ότι είναι αναγκαία και συγκεκριμένα έχει αποδειχτεί ότι οι αλγόριθμοι Μέγιστης Εντροπίας (Max Entropy) και Support Vector Machines μπορούν να επωφεληθούν από την εισαγωγή της ουδέτερης κλάσης και να βελτιώσουν τη συνολική ευστοχία των προβλέψεων.

Άλλες κατευθύνσεις που ερευνώνται είναι η αναγνώριση υποκειμενικότητας ή αντικειμενικότητας, και η κατηγοριοποίηση της άποψης για το αντικείμενο που πραγματεύεται το κείμενο σε επίπεδο γνωρισμάτων αυτού. Παράδειγμα για το πρώτο είναι η αναφορά σε λόγια τρίτου στο κείμενο (quoting) που δεν αφορούν αναγκαστικά την άποψη του συγγραφέα και για το δεύτερο η ανάλυση μίας κριτικής κάμερας στην οποία κατηγοριοποιούμε την γνώμη του συγγραφέα για το φακό, για την εικόνα, την αντοχή, την εστίαση κ.ο.κ.

Αμφότερες μπορούν να χρησιμοποιηθούν για τη βελτίωση και σωστή λειτουργία συγκεκριμένων μεθόδων και αλγορίθμων.

2.2.3 Ανάλυση κειμένου ως "σάκο από λέξεις"

Το μοντέλο σάκου από λέξεις (bag-of-words model) είναι μία απλή αναπαράσταση κειμένου που χρησιμοποιείται συχνά σε διάφορες εφαρμογές Επεξεργασίας Φυσικής Γλώσσας και όχι μόνο. Σε αυτή την αναπαράσταση το κείμενο είναι ένας "σάκος" ο οποίος περιέχει όλες τις λέξεις του κειμένου χωρίς να ενδιαφέρεται για γραμματική, συντακτικό, στίξη και ούτε καν τη σειρά τους στο κείμενο (μονάχα, στη γενική του μορφή, διατηρεί τις επαναλαμβανόμενες λέξεις).

Είναι βασική αναπαράσταση που χρησιμοποιείται για Κατηγοριοποίηση Κειμένων και ένα από τα πιο διαδεδομένα παραδείγματα είναι το spam filtering με χρήση αλγόριθμου Naïve Bayes.

2.3 Naive Bayes

Το πιο συνηθισμένο και απλό, πιθανώς, μοντέλο που χρησιμοποιείται στην ΚΒΑ είναι το απλοϊκό μοντέλο Bayes (Naive Bayes model). Το μοντέλο Naive Bayes κάνει την υπόθεση ότι τα χαρακτηριστικά (features, λέξεις στην περίπτωση μας) είναι ανεξάρτητα μεταξύ τους. Γι αυτό και για κείμενα μπορούμε να χρησιμοποιήσουμε το μοντέλο σάκου από λέξεις και να εξετάσουμε κάθε λέξη σαν ένα ξεχωριστό feature ανεξάρτητο από όλα τα άλλα.

Έστω ένα παράδειγμα E, με διάνυσμα χαρακτηριστικών

$$X = (x_1, x_2, ..., x_n).$$

Αυτό το παράδειγμα πρέπει να κατηγοριοποιηθεί σε μια κατηγορία c από ένα σύνολο κατηγοριών C. Εν προκειμένω το σύνολο C θα περιέχει μόνο δύο κατηγορίες, την αρνητική και τη θετική γνώμη, οπότε έστω $C = \{A, \Theta\}$. Η πιθανότητα το παράδειγμα να ανήκει στην κατηγορία c, δεδομένων των χαρακτηριστικών X, δίνεται από τον κανόνα του Bayes:

$$P(C|X) = \frac{P(X|C) \times P(C)}{P(X)}$$

δηλαδή, η πιθανότητα το παράδειγμα που μας παρουσιάζεται, με διάνυσμα χαρακτηριστικών , να ανήκει στην κατηγορία c—$P(c|X)$— είναι ίση με την πιθανότητα ένα τυχαίο διάνυσμα να ανήκει στην κατηγορία αυτή—$P(c)$, επί την πιθανότητα το διάνυσμα αυτό να εμφανιζόταν σε ένα παράδειγμα αυτής της κατηγορίας—$P(X|c)$, δια την πιθανότητα του διανύσματος αυτού να εμφανιστεί γενικά—$P(X)$.

Στην ορολογία των πιθανοτήτων, η πιθανότητα $P(c)$ ονομάζεται prior (εκ των προτέρων πιθανότητα), δηλαδή η πιθανότητα ένα παράδειγμα να ανήκει σε μια κατηγορία, όταν δε γνωρίζουμε καν το περιεχόμενό του (δηλαδή και το διάνυσμα που το αναπαριστά). Με τη γνώση του διανύσματος X, η εκ των προτέρων πιθανότητα μεταβάλλεται στην εκ των υστέρων πιθανότητα (posterior) $P(c|X)$. Αυτή είναι ίση με $P(c)$ επί τον λόγο $\frac{P(X|c)}{P(X)}$.

Όσον αφορά την τιμή $P(c)$, αυτή μπορεί θεωρητικά να επιλεχθεί βάσει του συνόλου εκπαίδευσης (π.χ. αν σε μια κατηγορία c ανήκει το 10% των παραδειγμάτων τότε $P(c) = 0.1$). Μια άλλη όμως επιλογή είναι να ισοκατανείμουμε τις πιθανότητες $P(c)$ στις διαθέσιμες κατηγορίες. Για την αρνητική/θετική γνώμη να επιλέξουμε δηλαδή $P(A) = P(\Theta) = 0.5$. Ο ταξινομητής που θα προκύψει καλείται "not biased", δηλαδή δεν είναι προκατειλημμένος απέναντι στην αρνητική ή τη θετική άποψη.

Οι τιμές $P(X|c)$ και $P(X)$ πρέπει να προσδιοριστούν από τα δεδομένα εκπαίδευσης. Επειδή σε πρώτη φάση οι πιθανότητες αυτές αναφέρονται σε μια συγκεκριμένη αναπαράσταση, π.χ. το διάνυσμα

$$X = \{\text{"έμεινα"} : 1, \text{"ευχαριστημένος"} : 1, \text{"πολύ"} : 1, \text{όλες οι άλλες λέξεις} : 0\}$$

που είναι μια bag-of-words αναπαράσταση της φράσης "Εμεινα πολύ ευχαριστημένος", θα είχαμε τιμές για τις δύο απαιτούμενες πιθανότητες μόνο εάν ένα παράδειγμα με την ίδια ακριβώς bag-of-words αναπαράσταση είχε εμφανιστεί στα δεδομένα εκπαίδευσης. Δεδομένου του μεγάλου αριθμού από διαφορετικές φράσεις ή κείμενα που οι άνθρωποι μπορούν να παράξουν, δεν μπορούμε να περιμένουμε ότι θα έχουμε συναντήσει την ίδια ακριβώς είσοδο στο σύνολο εκπαίδευσης.

Ο προσδιορισμός τιμών για τις πιθανότητες $P(X|c)$ διευκολύνεται με την υπόθεση ότι η τιμή κάθε χαρακτηριστικού του διανύσματος X είναι ανεξάρτητη από τις τιμές των άλλων, υπό τη συνθήκη c. Έτσι, η πιθανότητα $P(X|c)$

γράφεται $P(X_1 = x_1|c) \cdot P(X_2 = x_2|c) \cdot ... \cdot P(X_n = x_n|c)$. Οι επιμέρους αυτές πιθανότητες μπορούν ρεαλιστικά να προσεγγιστούν από τις αντίστοιχες συχνότητες των χαρακτηριστικών στα δεδομένα εκπαίδευσης.

2.4 Hidden Markov Model

Το δεύτερο μοντέλο που εφαρμόσαμε είναι το Κρυφό Μοντέλο Markov (Hidden Markov Model), και πιο συγκεκριμένα μια παραλλαγή που ονομάζεται Lexicalized Hidden Markov Model Integrating Part-of-Speech, που χρησιμοποιήθηκε στο σύστημα ανάλυσης άποψης OpinionMiner [JHS09]. Για να περιγράψουμε το σύστημα αυτό, πρώτα θα πούμε λίγα λόγια για τη θεωρία του Hidden Markov Model, του οποίου επέκταση είναι το σύστημα που υλοποιήσαμε.

2.4.1 Αλυσίδα Markov

Μια αλυσίδα Markov (Markov chain) είναι ένα πιθανοτικό μοντέλο το οποίο μεταβαίνει από κατάσταση σε κατάσταση, από ένα πεπερασμένο σύνολο πιθανών καταστάσεων. Το βασικότερο χαρακτηριστικό της είναι ότι το σε ποια κατάσταση θα μεταβεί το σύστημα εξαρτάται μόνο από την τωρινή κατάσταση, και όχι από την προηγούμενη αλληλουχία καταστάσεων που αυτό διένυσε, ιδιότητα που καλείται έλλειψη μνήμης (memorylessness). Η μετάβαση από κάθε κατάσταση στην επόμενη γίνεται βάσει των πιθανοτήτων μετάβασης, οι οποίες είναι διαφορετικές για κάθε κατάσταση.

Το σύστημα επίσης παράγει μία έξοδο, ανάμεσα στις μεταβάσεις του, από ένα σύνολο συμβόλων εξόδου, η κατανομή πιθανότητας των οποίων είναι διαφορετική για κάθε κατάσταση.

Το είδος του μοντέλου αυτού θα γίνει πιο κατανοητό με ένα παράδειγμα.

Παράδειγμα αλυσίδας Markov

Έστω σύνολο καταστάσεων $S = \{A, B, C\}$. Έστω οι αρχικές πιθανότητες (initial probabilities) $P(X_0 = i)$ που προσδιορίζουν τις πιθανότητες η πρώτη κατάσταση του συστήματος να είναι η i.

Κατάσταση	A	B	C
Αρχική Πιθανότητα	0.3	0.3	0.4

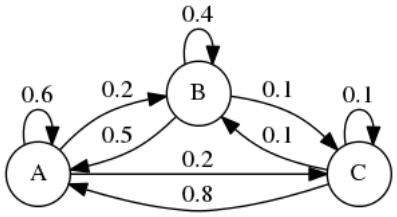
Έστω επίσης ο παρακάτω πίνακας μεταβάσεων $P(X \to Y)$ όπου στη θέση i, j του πίνακα βρίσκεται η πιθανότητα το σύστημα να μεταβεί από την κατάσταση i στην κατάσταση j.

$P(A \to A) = 0.6$	$P(A \to B) = 0.2$	$P(A \to C) = 0.2$
$P(B \to A) = 0.5$	$P(B \to B) = 0.4$	$P(B \to C) = 0.1$
$P(C \to A) = 0.8$	$P(C \to B) = 0.1$	$P(C \to C) = 0.1$

Υποθέτουμε επίσης ότι το σύνολο συμβόλων εξόδου είναι το $T = \{x, y\}$. Η κατανομή πιθανότητας των εξόδων ανά κατάσταση δίνεται από τον παρακάτω πίνακα:

Κατάσταση	Πιθανότητα εξόδου 'x'	Πιθανότητα εξόδου 'y'
A	0.1	0.9
B	0.5	0.5
C	0.3	0.7

Οι πιθανότητες μετάβασης από κατάσταση σε κατάσταση μπορούν να απεικονιστούν όπως στο σχήμα 2.1.



Σχήμα 2.1: Διάγραμμα μεταβάσεων αλυσίδας Markov

Η αλυσίδα Markov συνήθως περιγράφεται χρησιμοποιώντας την έννοια της χρονικής εξέλιξης, όπου οι μεταβάσεις πραγματοποιούνται σε διακριτές χρονικές στιγμές. Στην εφαρμογή του μοντέλου σε κείμενο, ο "διακριτός χρόνος" $t = 0, 1, 2, 3, \ldots$ αντιστοιχεί στην πρώτη, δεύτερη, τρίτη λέξη του κειμένου κ.ο.κ. και οι έξοδοι του συστήματος είναι οι ίδιες οι λέξεις.

2.4.2 Το Κρυφό Μοντέλο Markov

Το Κρυφό Μοντέλο Markov (Hidden Markov Model, εν συντομία HMM) αναφέρεται σε ένα μοντέλο Markov, του οποίου οι καταστάσεις είναι κρυφές, ενώ παρατηρήσιμη είναι μόνο η αλληλουχία των εξόδων. Τα HMMs βρίσκουν εφαρμογή σε αρκετά σημαντικά προβλήματα μηχανικής μάθησης, όπως η αναγνώριση φωνής (speech recognition), η αναγνώριση χειρόγραφων κειμένων (handwriting recognition), η αναγνώριση χειρονομιών (gesture recognition) και η επισήμανση μέρους του λόγου (part-of-speech tagging).

Για να χρησιμοποιηθεί ένα κρυφό μοντέλο Markov στη μηχανική μάθηση γίνεται η υπόθεση ότι τα υπό εξέταση δεδομένα έχουν παραχθεί από ένα τέτοιο μοντέλο και - κατά την εκπαίδευση - με βάση αυτά προσπαθούμε να

προσεγγίσουμε τις παραμέτρους του μοντέλου (πιθανότητες μετάβασης καταστάσεων, πιθανότητες εξόδων). Αυτό το επιτυγχάνουμε διότι στα δεδομένα εκπαίδευσης μας είναι γνωστή η αλληλουχία των καταστάσεων – τα δεδομένα είναι επισημειωμένα (tagged) με την κατάσταση του μοντέλου που παρήγαγε κάθε σύμβολο εξόδου.

Όταν στη συνέχεια το εκπαιδευμένο κρυφό μοντέλο Markov εφαρμόζεται σε μη επισημειωμένα δεδομένα, συμπεραίνεται η πιο πιθανή αλληλουχία κρυφών καταστάσεων την οποία το μοντέλο διένυσε, και οι κρυφές καταστάσεις αυτές αποτελούν την πληροφορία που το σύστημα τελικά αποδίδει στο χρήστη του.

Σημειώνεται ότι η επιλογή του συνόλου S των δυνατών καταστάσεων του μοντέλου είναι μια μεταπαράμετρος που επιλέγεται από τον ερευνητή εκ των προτέρων, με στόχο να ταιριάζει στη φύση του προβλήματος που η μηχανική μάθηση του μοντέλου σκοπεύει να λύσει. Αυτό το βήμα δεν είναι σε καμία περίπτωση μια γρήγορη ή αυτονόητη επιλογή. Για παράδειγμα, όταν το Κρυφό Μοντέλο Markov χρησιμοποιείται ως επισημειωτής μέρους του λόγου (part-of-speech tagger), τότε οι κρυφές καταστάσεις είναι τα ίδια τα μέρη του λόγου. Όμως, το ακριβές σύνολο των μερών του λόγου που θα επιλεχθεί εξαρτάται από τους σκοπούς του συστήματος, και εν προκειμένω σχετίζεται και με την επιστήμη της Γλωσσολογίας. Ένα απλό σύνολο καταστάσεων – μερών του λόγου για την ελληνική γλώσσα θα μπορούσαν να είναι τα γνωστά 10 μέρη του λόγου (Άρθρο, Ουσιαστικό, Επίθετο, κ.λ.π.) (Γραμματική Μανόλη Τριανταφυλλίδη). Όμως μπορεί να κριθεί σκόπιμο οι κατηγορίες αυτές να είναι πιο ειδικές, για παράδειγμα "Οριστικό Άρθρο Γένους Θηλυκού Ονομαστική Ενικού", "Οριστικό Άρθρο Γένους Θηλυκού Γενική Ενικού", κ.λ.π. Έτσι, το προτεινόμενο σύνολο ετικετών μπορεί να ξεπεράσει σε μέγεθος μέχρι και τις 100 ετικέτες (καταστάσεις), ενώ στην περίπτωση του συνόλου που χρησιμοποιείται από το Ινστιτούτο Επεξεργασίας του Λόγου για τα ελληνικά, το ILSP-PAROLE, διακρίνονται 584 ετικέτες. Συνεπώς, το σύνολο των κρυφών καταστάσεων του μοντέλου Markov επιλέγεται βάσει του προβλήματος προς λύση, και προφανώς καθορίζει το προκύπτον σύστημα σε πολύ μεγάλο βαθμό [1].

Επίσης, εκτός από την αποτελεσματικότητα/ορθότητα, η επιλογή του συνόλου αυτού επηρεάζει τη χρονική πολυπλοκότητα της εκτέλεσης κυρίως της φάσης της εφαρμογής του εκπαιδευμένου μοντέλου, όπως θα δούμε στην επόμενη υποενότητα (αλγόριθμος Viterbi).

Υπάρχουν πάντως μέθοδοι με βάση τις οποίες το σύνολο των καταστάσεων του μοντέλου Markov μπορεί να μεταβάλλεται κατά τη διάρκεια της εκπαίδευσης (για παράδειγμα, καταστάσεις που πλεονάζουν να αφαιρούνται ή νέες να προστίθενται). Τέτοιες μέθοδοι δεν χρησιμοποιήθηκαν στη δικιά μας εφαρμογή του Κρυφού Μοντέλου Markov, αφού οι καταστάσεις επιλέχθηκαν εκ των

[1] Η αναφορά στο part-of-speech tagging δεν γίνεται μόνο για να σχολιάσουμε τη διαδικασία επιλογής του συνόλου κρυφών καταστάσεων σ' ένα HMM, αλλά και γιατί το μέρος του λόγου ενσωματώνεται στο συγκεκριμένο είδος HMM που υλοποιήσαμε, όπως θα δούμε παρακάτω.

προτέρων για να ταιριάζουν στο aspect-oriented sentiment analysis και μάλιστα ειδικά στο domain των κριτικών ξενοδοχείων, όπως θα δούμε παρακάτω αναλυτικότερα.

Ο πίνακας 2.1 περιγράφει πως βλέπουμε τα διάφορα μέρη του συστήματος HMM στις φάσεις της εκπαίδευσης και της εφαρμογής του μοντέλου.

Φάση	Έξοδοι (tokens)	Καταστάσεις (states)	Πιθανότητες μεταβάσεων και εξόδων (παράμετροι)	Αλγόριθμος
Εκπαίδευση	Γνωστές	Γνωστές	Άγνωστες (μαθαίνονται)	Maximum Likelihood Estimation
Εφαρμογή	Γνωστές	Άγνωστες (συμπεραίνονται)	Γνωστές	Viterbi

Πίνακας 2.1: Εκπαίδευση και Εφαρμογή Κρυφού μοντέλου Markov

2.4.3 Εκπαίδευση Hidden Markov Model – Maximum Likelihood Estimation

Στο επισημειωμένο training set, σε κάθε token (για ανάλυση κειμένου αυτά περίπου αντιστοιχούν σε λέξεις) έχει προσδιοριστεί η κρυφή κατάσταση του συστήματος Markov που το παρήγαγε. Η διαδικασία αυτή αναγκαστικά γίνεται χειροκίνητα, όπου με τη βοήθεια συνήθως κάποιου απλού εργαλείου με γραφική διεπαφή (GUI) επιλέγονται οι λέξεις και οι αντίστοιχες καταστάσεις.

Έχοντας έτοιμο το training set, το ζητούμενο είναι να προσδιοριστούν οι παράμετροι του μοντέλου, δηλαδή οι πιθανότητες μετάβασης από κατάσταση σε κατάσταση, η κατανομή πιθανότητας των εξόδων ανά κατάσταση, αλλά και οι αρχικές πιθανότητες (σε ποια κατάσταση βρίσκεται το σύστημα στην πρώτη λέξη). Η πιο βασική προσέγγιση είναι η Εκτίμηση Μέγιστης Πιθανοφάνειας (Maximum Likelihood Estimation). Αυτό σημαίνει ότι επιλέγουμε τις τιμές των πιθανοτικών παραμέτρων του συστήματος ούτως ώστε η πιθανοφάνεια των δεδομένων του training set να μεγιστοποιείται.

Αποδεικνύεται ότι οι τιμές αυτές προκύπτουν από τις αντίστοιχες σχετικές συχνότητες στα δεδομένα εκπαίδευσης. Ένα απλό παράδειγμα δείχνει πολύ ξεκάθαρα πως εφαρμόζεται αυτή η μέθοδος:

Έστω ένα Κρυφό Μοντέλο Markov με καταστάσεις $S = \{A, B\}$ και αλφάβητο εξόδου $T = \{x, y\}$. Εάν τα δεδομένα εκπαίδευσης περιείχαν την ακόλουθη αλληλουχία καταστάσεων (αγνοώντας προσωρινά τις εξόδους):

$$A \to A \to A \to A \to A \to A \to B \to B \to A \to$$
$$\to A \to A \to A \to B \to A \to B \to B \to B \to A \to A$$

Φαίνεται ότι υπάρχουν 12 μεταβάσεις που ξεκινούν από Α, απ' τις οποίες 9 οδηγούν σε Α ξανά, ενώ 3 οδηγούν σε Β. Από Β ξεκινούν 6 μεταβάσεις, από τις οποίες 3 οδηγούν σε Α και 3 σε Β ξανά. Η ανάθεση των πιθανοτήτων μεταβάσεων $P(A \to A)$, $P(A \to B)$, $P(B \to A)$ και $P(B \to B)$ που μεγιστοποιεί την πιθανοφάνεια της παρατηρηθείσας ακολουθίας είναι αυτή των σχετικών συχνοτήτων αυτών των μεταβάσεων, δηλαδή:

$$\begin{aligned} P(A \to A) &= 9/12 &&= 0.75 \\ P(A \to B) &= 3/12 &&= 0.25 \\ P(B \to A) &= 3/6 &&= 0.5 \\ P(B \to B) &= 3/6 &&= 0.5 \end{aligned}$$

Αναλόγως χειριζόμαστε και τις πιθανότητες εξόδου. Αν στο προηγούμενο παράδειγμα προσθέσουμε και τις εξόδους με το συμβολισμό $A(x)$ να σημαίνει ότι έχουμε έξοδο x από κατάσταση A:

$$A(x) \to A(x) \to A(y) \to A(x) \to A(x) \to A(y) \to B(y) \to$$
$$\to B(y) \to A(x) \to A(y) \to A(x) \to A(y) \to B(x) \to$$
$$\to A(y) \to B(y) \to B(x) \to B(y) \to A(x) \to A(y)$$

Τότε η ανάθεση των πιθανοτήτων εξόδου που μεγιστοποιεί την πιθανοφάνεια του παραπάνω αποτελέσματος, πάλι βάσει σχετικών συχνοτήτων, θα ήταν:

$$\begin{aligned} P(x|A) &= 7/13 &&= 0.54 \\ P(y|A) &= 6/13 &&= 0.46 \\ P(x|B) &= 2/6 &&= 0.33 \\ P(y|B) &= 4/6 &&= 0.67 \end{aligned}$$

Τέλος, οι μόνες πιθανότητες που μένει να προσδιοριστούν είναι οι αρχικές πιθανότητες, δηλαδή η πιθανότητα η πρώτη κατάσταση να είναι Α ή Β. Στο παράδειγμα έχουμε μόνο μια ακολουθία, οπότε αναγκαστικά θα είχαμε $P(A) = 1$ και $P(B) = 0$. Σε πραγματικά δεδομένα εκπαίδευσης θα είχαμε πολλά παραδείγματα, οπότε οι πιθανότητες αυτές θα προσεγγίζονταν από τις σχετικές συχνότητες. Εάν όμως πιθανότητες με μηδενική τιμή είναι ανεπιθύμητες (γιατί μηδενική πιθανότητα σε μια μετάβαση ή έξοδο καθιστά αυτήν αδύνατη, ενώ μπορεί να είναι απλά σπάνια και να μη συναντήθηκε στα δεδομένα εκπαίδευσης), μπορεί να εφαρμοστεί η μέθοδος της Εξομάλυνσης Laplace (Laplace smoothing ή add-one smoothing). Σε αυτήν ο τύπος της σχετικής συχνότητας μεταβάλλεται ώστε να είναι αδύνατο να μηδενιστεί. Δηλαδή:

Αντί για

$$P(A \to A) = \frac{C(A \to A)}{\sum_{s \in S} C(A \to s)}$$

και

$$P(A \to B) = \frac{C(A \to B)}{\sum_{s \in S} C(A \to s)}$$

που υπολογίστηκαν προηγουμένως, υπολογίζονται τώρα ως:

$$P(A \to A) = \frac{1 + C(A \to A)}{|S| + \sum_{s \in S} C(A \to s)}$$

και

$$P(A \to B) = \frac{1 + C(A \to B)}{|S| + \sum_{s \in S} C(A \to s)}$$

όπου $|S|$ το πλήθος των καταστάσεων και $C(s_i \to s_j)$ το πλήθος των μεταβάσεων από κατάσταση s_i σε κατάσταση s_j. Έτσι οι πιθανότητες μετάβασης από Α προς όλες τις πιθανές επόμενες καταστάσεις συνεχίζουν να έχουν άθροισμα 1, αλλά τώρα καμία δε γίνεται να είναι μηδενική, αλλά παίρνει μια ελάχιστη τιμή.

Η Εξομάλυνση Laplace μπορεί να εφαρμοστεί σε όλα τα είδη πιθανοτήτων που υπολογίζονται κατ' αυτόν τον τρόπο, εκτός εάν βάσει λόγων σχεδιασμού του μοντέλου κάποιες αρχικές καταστάσεις ή μεταβάσεις κρίνεται σκόπιμο να απαγορευτούν. Η Εξομάλυνση Laplace μπορεί να εφαρμοστεί αναλόγως και στις πιθανότητες των εξόδων.

Αφού ολοκληρώσουμε την εκπαίδευση έχουμε ένα μοντέλο Markov με γνωστές όλες του τις παραμέτρους και με βάση αυτό προσπαθούμε να βρούμε την πιο πιθανή αλληλουχία των (κρυφών) καταστάσεων σε μη ταξινομημένα δεδομένα. Η τελευταία αυτή διαδικασία επιτυγχάνεται με τον αλγόριθμο Viterbi.

2.4.4 Εφαρμογή Hidden Markov Model – Αλγόριθμος Viterbi

Όπως είπαμε, το πρόβλημα που λύνει ο αλγόριθμος Viterbi είναι αυτό του προσδιορισμού των πιο πιθανών κρυφών καταστάσεων από τις οποίες πέρασε το μοντέλο Markov, δεδομένων των εξόδων του συστήματος. Επίσης δεδομένες θεωρούνται όλες οι πιθανοτικές παράμετροι του συστήματος (έχουν ήδη προσδιοριστεί στη φάση της εκπαίδευσης).

Ο αλγόριθμος έχει πολυπλοκότητα $(N \cdot S^2)$ όπου N το μήκος της εισόδου σε tokens και S ο αριθμός των κρυφών καταστάσεων. Επειδή η πολυπλοκότητα είναι τετραγωνική με τον αριθμό των καταστάσεων, προκύπτει ότι έχει σημασία να δίνεται προσοχή στο πλήθος των καταστάσεων για να αποφεύγεται μεγάλο χρονικό κόστος.

Ο αλγόριθμος Viterbi είναι μια περίπτωση της κατηγορίας αλγορίθμων του δυναμικού προγραμματισμού, και όπως όλοι αυτοί οι αλγόριθμοι, χαρακτηρίζεται από μια αναδρομική σχέση βέλτιστης υπο-δομής.

Αν υποθέσουμε ότι παρατηρούμε εξόδους $x_1, x_2, \ldots, x_n$, αναζητούμε εκείνη την αλληλουχία κρυφών καταστάσεων $s_1, s_2, \ldots, s_n$ για την οποία η πιθανότητα $P(s_1, s_2, \ldots, s_n | x_1, x_2, \ldots, x_n)$ γίνεται μέγιστη.

Συμβολίζοντας επίσης:

- Την αρχική πιθανότητα για την κατάσταση s_i με $P(S_1 = s_i)$
- Την πιθανότητα μετάβασης από την κατάσταση s_i στην κατάσταση s_j με $P(s_i \rightarrow s_j)$
- Την πιθανότητα εξόδου x_k υπό την κατάσταση s_i με $P(x_k | s_i)$
- και, κυριότερα, με $V_{t,k}$ την πιθανότητα της πιο πιθανής αλληλουχίας καταστάσεων μήκους t στην οποία η t-οστή κατάσταση είναι η s_k (δηλαδή $S_t = s_k$)

έχουμε

$$\begin{aligned} V_{1,k} &= P(S_1 = s_k) \cdot P(x_1 | s_k) \\ V_{t,k} &= \max_{s_i \in S} \left\{ V_{(t-1),i} \cdot P(s_i \rightarrow s_k) \cdot P(x_t | s_k) \right\} \end{aligned}$$

Με βάση αυτές τις δύο σχέσεις μπορεί να υλοποιηθεί ο bottom-up υπολογισμός των ποσοτήτων $V_{t,k}$ μέχρις ότου να φτάσουμε στα $V_{n,k}$, $\forall s_k \in S$. Η μέγιστη αυτών των $|S|$ πιθανοτήτων είναι η πιθανότητα της πιο πιθανής ακολουθίας $s_1, s_2, \ldots, s_n$ κρυφών καταστάσεων. Για να ξέρουμε και την ίδια την πιο πιθανή ακολουθία καταστάσεων, απαιτείται να ενημερώνουμε έναν πίνακα από $|S|$ λίστες, που περιέχει τις πιθανότερες ακολουθίες μήκους t σε κάθε βήμα του αλγορίθμου, που καταλήγουν σε καθεμιά από τις καταστάσεις του συνόλου S.

2.4.5 Lexicalized HMM Integrating Part-of-Speech

Αφού βάλαμε κάποιες βάσεις για τη χρήση των Κρυφών Μοντέλων Markov στη Μηχανική Μάθηση γενικά, θα περιγράψουμε τις τροποποιήσεις που εισαγάγει το μοντέλο Lexicalized Hidden Markov Model Integrating POS. Οι πιθανότητες μετάβασης και παραγωγής εξόδων εμπλουτίζονται από εξαρτήσεις από το μέρος του λόγου των λέξεων ("integrating POS") και από τις προηγούμενες λέξεις ("lexicalized").

Μια διαφοροποίηση στην ορολογία είναι ότι οι κρυφές καταστάσεις του μοντέλου θα καλούνται ετικέτες (Tags) και θα συμβολίζονται πλέον με το γράμμα t. Ο λόγος είναι ότι αυτές νοούνται ως ετικέτες που σημειώνονται στα δεδομένα εκπαίδευσης, ή ως ετικέτες που ανατίθενται από το μοντέλο στα άγνωστα δεδομένα, λέξη προς λέξη. Οι ετικέτες αποτελούν λοιπόν την κρυφή κατάσταση.

Algorithm 1 Αλγόριθμος Viterbi

```
1: function VITERBI(S, xs, Ptrans, Pinit, Pout)
2:     V ← {}                                          ▷ {}: Associative Arrays
3:     Paths ← {}
4:     V' ← {}
5:     Paths' ← {}
6:     for all s ∈ S do
7:         V[s] = Pinit[s] · Pout[s][xs[1]]
8:         Paths[s] = [ ]                              ▷ [ ]: Άδεια λίστα
9:     end for
10:    i ← 2
11:    while i < |xs| do
12:        x ← xs[i]
13:        for all s ∈ S do
14:            s_choice ← Argmax_{s'∈S}(V[s'] · Ptrans[s', s])
15:            Paths'[s] ← Paths[s] + [s_choice]        ▷ +: Συνένωση λιστών
16:            V'[s] ← V[s_choice] · Ptrans[s_choice, s] · Pout[s][x]
17:        end for
18:        swap(V, V')
19:        swap(Paths, Paths')
20:        i ← i + 1
21:    end while
22:    s_ret ← Argmax_{s∈S}(V[s])
23:    path_ret ← Paths[s_ret]
24:    return path_ret
25: end function
```

Τα άλλα δύο δεδομένα, το μέρος του λόγου και η λέξη, αποτελούν τη "φανερή κατάσταση" του μοντέλου. Το μέρος του λόγου (part of Speech) συμβολίζεται με s ενώ η λέξη (Word) με w. Το κείμενο τελικά παρουσιάζεται στη φάση της εκπαίδευσης ως μια ακολουθία από τριάδες t_i, s_i, w_i, ενώ στη φάση της αποκωδικοποίησης άγνωστου κειμένου λείπουν τα t_i και θεωρούνται δεδομένα τα s_i, w_i.

Τελικά, το μοντέλο μεταβάσεων του HMM επεκτείνεται ώστε:

- Η πιθανότητα μετάβασης στην τωρινή κατάσταση t_n εξαρτάται όχι μόνο από την προηγούμενή κατάσταση t_{n-1}, αλλά και από την προηγούμενη λέξη w_{n-1}.
- Η πιθανότητα εμφάνισης του τωρινού μέρους του λόγου s_n εξαρτάται από την τωρινή κατάσταση t_n και την προηγούμενη λέξη w_{n-1}.
- Η πιθανότητα εμφάνισης της τωρινής λέξης w_n εξαρτάται όχι μόνο από την τωρινή κατάσταση t_n, αλλά και από το τωρινό μέρος του λόγου s_n και την προηγούμενη λέξη w_{n-1}.

Δηλαδή οι κατά συνθήκη πιθανότητες που το ορίζουν είναι οι ακόλουθες:

- $P(t_n|t_{n-1}, w_{n-1})$
- $P(s_n|t_n, w_{n-1})$
- $P(w_n|t_n, s_n, w_{n-1})$

και φυσικά για την πληρότητα του μοντέλου χρειάζονται και οι αρχικές πιθανότητες:

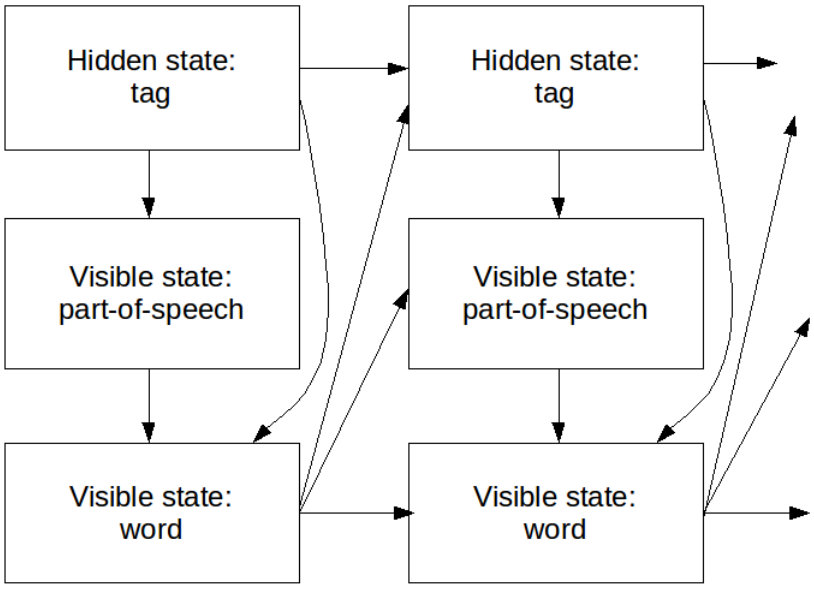
- $P(t_0)$
- $P(s_0|t_0)$
- $P(w_0|t_0, s_0)$

Οι εξαρτήσεις μεταξύ καταστάσεων, μέρους του λόγου και λέξεων παρουσιάζονται και στο σχήμα 2.2.

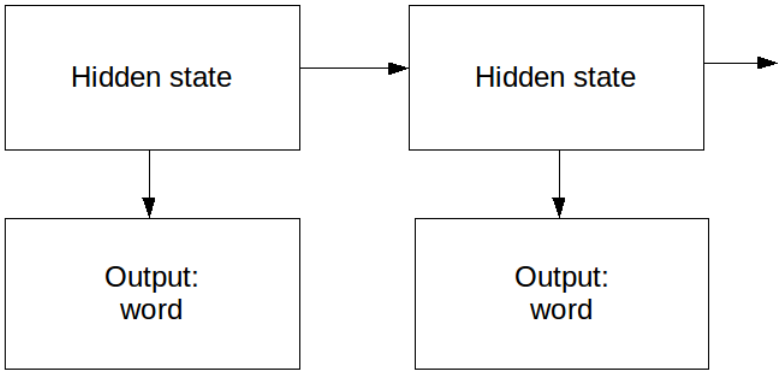
Για λόγους σύγκρισης, ακολουθεί και το σχήμα 2.3 με το απλό Hidden Markov Model.

Με το πιο σύνθετο μοντέλο των πιθανοτικών εξαρτήσεων είναι δυνατό να γίνει αναγνώριση φράσεων βάσει του μέρους του λόγου ή και συγκεκριμένων λέξεων.

Για να γίνει πιο κατανοητή η λειτουργία αυτού του μοντέλου, πρέπει φυσικά να παραθέσουμε το σύνολο των κρυφών καταστάσεων που επιλέχθηκε, το τι σηματοδοτούν αυτές και το πως τα δεδομένα εκπαίδευσης επισημειώθηκαν βάσει αυτών.



Σχήμα 2.2: Εξαρτήσεις μεταβάσεων στο Lexicalized Hidden Markov Model Integrating Part-of-Speech



Σχήμα 2.3: Εξαρτήσεις μεταβάσεων στο απλό Hidden Markov Model

2.4.6 Το σύστημα OpinionMiner

Στο σύστημα OpinionMiner, στο οποίο βασίζεται το δικό μας σύστημα, εφαρμόστηκε σε κριτικές προϊόντων σε ηλεκτρονικό κατάστημα (Amazon) και συγκεκριμένα σε κριτικές φωτογραφικών μηχανών. Ο σκοπός του συστήματος ήταν να εξάγονται θετικές ή αρνητικές κρίσεις για συγκεκριμένες εκφάνσεις (aspects) των προϊόντων Έτσι, το σύνολο των κρυφών καταστάσεων (που στη δημοσίευση αναφέρονται ως "ετικέτες" - tags) που ορίστηκαν ήταν το εξής (πίνακας 2.2):

Ετικέτα	**Οντότητες που αντιπροσωπεύει**
PROD_FEAT	Χαρακτηριστικό (feature) του προϊόντος
PROD_PARTS	Εξάρτημα (part) του προϊόντος
PROD_FUNCTION	Λειτουργία (function) του προϊόντος
OPINION_POS_EXP	Άμεση (explicit) θετική γνώμη
OPINION_NEG_EXP	Άμεση (expilict) αρνητική γνώμη
OPINION_POS_IMP	Έμμεση (implicit) θετική γνώμη
OPINION_NEG_IMP	Έμμεση (implicit) αρνητική γνώμη
BG	Οι υπόλοιπες λέξεις (background words)

Πίνακας 2.2: Βασικές ετικέτες OpinionMiner

Με βάση αυτό το σύνολο ετικετών, το κείμενο των κριτικών στο σύνολο εκπαίδευσης θα επισημειωνόταν χειρωνακτικά από τους ερευνητές, και ανάλογες ετικέτες θα επισημαίνονταν αυτόματα από το σύστημα στα άγνωστα κείμενα που του υποβάλλονταν στη φάση των δοκιμών.

Για παράδειγμα, η φράση "The lens is quite impressive." θα επισημειωνόταν, λέξη προς λέξη, ως εξής:

The	lens	is	quite	impressive	.
BG	PROD_PART	BG	BG	OPINION_POS_EXP	BG

Όμως οι ετικέτες αυτές δεν εφαρμόζονται μόνο σε μεμονωμένες λέξεις, αλλά και σε μικρές φράσεις. Για παράδειγμα η φράση "I love the ease of transfering the pictures to my computer." θα επισημειωνόταν όπως παρακάτω, θεωρώντας ότι η φράση "ease of transfering the pictures" είναι ένα "feature" του προϊόντος:

I	love	the	ease	of	transferring	the	pictures
BG	OPINION_POS_EXP	BG	PROD_FEAT				

to	my	computer	.
BG	BG	BG	BG

Επειδή στο Hidden Markov Model, κάθε λέξη είναι μια έξοδος και παράγεται από μία μόνο κατάσταση, και για να διευκολυνθεί η αναγνώριση αυτών των μικρών φράσεων, εάν μια "ετικέτα" εφαρμοστεί σε μια φράση μήκους πάνω από μια λέξη, χρησιμοποιούνται οι παρακάτω σύνθετες ετικέτες, οι οποίες παράγονται από τις βασικές προσθέτοντας τις καταλήξεις "-BOE", "-MOE" και "-EOE" που αντίστοιχα σημαίνουν "Beginning of Entity", "Middle of Entity" και "End of Entity". Οπότε αφού σημειωθεί χειρονακτικά από τον ερευνητή η παραπάνω φράση, οι ετικέτες θα μεταβληθούν αυτόματα σε:

I	love	the	ease	of
BG	OPINION_POS_EXP	BG	PROD_FEAT-BOE	PROD_FEAT-MOE

transferring	the	pictures	to
PROD_FEAT-MOE	PROD_FEAT-MOE	PROD_FEAT-EOE	BG

my	computer	.
BG	BG	BG

Έτσι, για καθεμιά από τις 7 βασικές ετικέτες (εκτός της BG) του πίνακα 2.2 υπάρχουν 4 τελικά ετικέτες, αυτές με τα τρία επιθέματα -BOE, -MOE και -EOE, αλλά και η ετικέτα χωρίς επίθεμα που χρησιμοποιείται για σημειωμένες οντότητες μήκους μιας λέξης. Καθεμιά από αυτές τις τελικές ετικέτες αντιστοιχεί σε μια κρυφή κατάσταση στο μοντέλο Markov. Το σύνολο των κρυφών καταστάσεων τελικά θα είχε $4 \cdot 7 + 1 = 29$ καταστάσεις.

2.4.7 Επιλογή ετικετών για το domain των ξενοδοχείων

Αναφέραμε προηγουμένως ότι η συζητούμενη εφαρμογή ενός Hidden Markov Model προϋπόθετε την επιλογή κατάλληλων ετικετών, που θα αντιστοιχούν στη θεματική περιοχή των δεδομένων όπου το μοντέλο θα εφαρμοστεί. Έτσι, επιλέξαμε μια σειρά από εκφάνσεις (aspects) οι οποίες σχολιάζονταν θετικά ή αρνητικά στις κριτικές ξενοδοχείων που είχαμε διαθέσιμες. Οι 5 εκφάνσεις του προϊόντος (ξενοδοχείου) που επιλέξαμε ήταν η εξυπηρέτηση (SERVICE), το προσωπικό (STAFF), οι εγκαταστάσεις (BUILDING), η τοποθεσία (LOCATION) και το κόστος (COST).

Επίσης, σε αντίθεση με το σύστημα OpinionMiner, δε χρησιμοποιήσαμε ετικέτες έμμεσης θετικής/αρνητικής γνώμης, παρά μόνο άμεσης γνώμης, με ονόματα απλώς POSITIVE και NEGATIVE.

Το σύνολο των ετικετών που χρησιμοποιήσαμε ακολουθεί στον πίνακα 2.3.

Για παράδειγμα, η φράση "πολύ μεγάλη εξυπηρέτηση από τους ιδιοκτήτες" που συναντήθηκε στο σύνολο δεδομένων μας σημειώθηκε ως εξής:

πολύ	μεγάλη	εξυπηρέτηση	από	τους	ιδιοκτήτες
BG	POSITIVE	SERVICE	BG	BG	STAFF

Ετικέτα	**Οντότητες που αντιπροσωπεύει**
SERVICE	Παρεχόμενες υπηρεσίες
STAFF	Προσωπικό του ξενοδοχείου
BUILDING	Κτιριακές εγκαταστάσεις
LOCATION	Τοποθεσία του ξενοδοχείου
COST	Κόστος
POSITIVE	Θετική γνώμη
NEGATIVE	Αρνητική γνώμη
BG	Οι υπόλοιπες λέξεις

Πίνακας 2.3: Βασικές ετικέτες στο domain των ξενοδοχείων

Η ακολουθία των ετικετών POSITIVE και SERVICE υποδηλώνει ότι ο σχολιαστής κρίνει θετικά την έκφανση SERVICE του ξενοδοχείου.

Το σύστημα τελικά δουλεύει ως εξής: μια ποσότητα δεδομένων επισημειώνεται χειρονακτικά κατ᾽ αυτόν τον τρόπο, το μοντέλο εκπαιδεύεται στα δεδομένα αυτά, και αφού εκπαιδευτεί, μπορεί να εφαρμοστεί σε νέα, άγνωστα δεδομένα, όπου και θα σημειώνει τις ετικέτες από μόνο του, εξάγοντας τελικά πληροφορία για τη θετική/αρνητική γνώμη για τη σχετική έκφανση. Το πως ποσοτικοποιήθηκε η απόδοση του μοντέλου στα δεδομένα μας αναφέρεται στο κεφάλαιο των πειραματικών αποτελεσμάτων, στην ενότητα 4.4.

Ο ρόλος του μέρους του λόγου

Όπως είπαμε παραπάνω, το μέρος του λόγου κάθε λέξης συνδέεται πιθανοτικά με την κρυφή κατάσταση—ετικέτα. Ο σκοπός αυτής της τροποποίησης του απλού HMM είναι να αναγνωρίζονται μοτίβα συσχέτισης ετικετών και μερών του λόγου, για παράδειγμα οι θετικές λέξεις είναι πολύ συχνά επίθετα, και συχνά ακολουθούνται από την έκφανση του ξενοδοχείου, που είναι με τη σειρά της συχνά ουσιαστικό, π.χ. “καθαρό δωμάτιο”, “ευγενικό προσωπικό” ή “ικανοποιητικό πρωινό”.

Ο ρόλος της εξάρτησης από λέξεις (lexicalized)

Η δεύτερη τροποποίηση στο απλό Hidden Markov Model ήταν η εξάρτηση της κρυφής κατάστασης από τις ίδιες τις λέξεις. Με αυτόν τον τρόπο επίσης μπορούν να αναγνωρίζονται μοτίβα, που αυτή τη φορά αφορούν συγκεκριμένες λέξεις. Για παράδειγμα, η λέξη “πολύ” έχει αυξημένη πιθανότητα να ακολουθείται από λέξη γνώμης π.χ. “πολύ ευγενικό προσωπικό”.

Εκπαίδευση μοντέλου—Maximum Likelihood Estimation

Η εκπαίδευση του μοντέλου γίνεται με το να δοθούν στις πιθανοτικές του παραμέτρους οι τιμές εκείνες που μεγιστοποιούν την πιθανοφάνεια των δεδομένων εκπαίδευσης. Έχουμε ήδη περιγράψει πως γίνεται αυτό σε ένα απλό Hidden Markov Model και εδώ θα δώσουμε τους τύπους της πιο σύνθετης περίπτωσης που ενσωματώνει λέξεις και μέρος του λόγου.

Οι πιθανότητες μετάβασης υπολογίζονται με βάση τις σχετικές συχνότητες εμφάνισής τους στα δεδομένα εκπαίδευσης:

- $P(t_n|t_{n-1}, w_{n-1}) = \frac{C(t_{n-1},w_{n-1},t_n)}{\sum_t C(t_{n-1},w_{n-1},t)}$
- $P(s_n|t_n, w_{n-1}) = \frac{C(w_{n-1},t_n,s_n)}{\sum_s C(w_{n-1},t_n,s)}$
- $P(w_n|t_n, s_n, w_{n-1}) = \frac{C(w_{n-1},t_n,s_n,w_n)}{\sum_w C(w_{n-1},t_n,s_n,w)}$

όπου $C(t_{n-1}, w_{n-1}, t_n)$ π.χ. σημαίνει το πλήθος των εμφανίσεων της κατάστασης t_n όταν προηγείται η κατάσταση t_{n-1} και η λέξη w_{n-1}. Σημειώνεται ότι στα κλάσματα αυτά, για τα αθροίσματα των παρονομαστών ισχύει π.χ. $\sum_t C(t_{n-1}, w_{n-1}, t) = C(t_{n-1}, w_{n-1})$, παρόμοια και για τα άλλα δύο.

Με βάση αυτόν τον τρόπο υπολογισμού των πιθανοτήτων δημιουργείται ένα θέμα. Όταν οι πιθανότητες μετάβασης υπολογίζονται και με βάση τη λέξη, είναι αναμενόμενο πολλοί συνδυασμοί να εντοπίζονται σπάνια (οπότε οι εκτιμήσεις να μην είναι πολύ αξιόπιστες), ή ακόμα και να εμφανιστούν άγνωστοι συνδυασμοί την ώρα της εφαρμογής του μοντέλου. Για να αποφευχθούν αυτές οι καταστάσεις, την ώρα της εκπαίδευσης μαθαίνονται και οι πιθανότητες μετάβασης αγνοώντας την προηγούμενη λέξη (non-lexicalized model), με βάση τις παρακάτω σχέσεις:

- $P(t_n|t_{n-1}) = \frac{C(t_{n-1},t_n)}{\sum_t C(t_{n-1},t)}$
- $P(s_n|t_n) = \frac{C(t_n,s_n)}{\sum_s C(t_n,s)}$
- $P(w_n|t_n, s_n) = \frac{C(t_n,s_n,w_n)}{\sum_w C(t_n,s_n,w)}$

και τη στιγμή της εφαρμογής του μοντέλου, η τελική πιθανότητα υπολογίζεται ως η γραμμική εξομάλυνση μεταξύ των δύο:

- $P'(t_n|t_{n-1}, w_{n-1}) = \lambda P(t_n|t_{n-1}, w_{n-1}) + (1 - \lambda)P(t_n|t_{n-1})$
- $P'(s_n|t_n, w_{n-1}) = \alpha P(s_n|t_n, w_{n-1}) + (1 - \alpha)P(s_n|t_n)$
- $P'(w_n|t_n, s_n, w_{n-1}) = \beta P(w_n|t_n, s_n, w_{n-1}) + (1 - \beta)P(w_n|t_n, s_n)$

όπου λ, α και β επιλέγονται από τον ερευνητή ως μεταπαράμετροι. Οι τιμές τους πρέπει να ανήκουν στο διάστημα $[0, 1]$, όπου με τιμή 0 το lexicalized στοιχείο αγνοείται, ενώ με τιμή 1 δεν υπάρχει εξομάλυνση.

Επίσης, όπως αναφέρθηκε και στην υποενότητα 2.4.3, στους υπολογισμούς των πιθανοτικών αυτών παραμέτρων μπορεί να εφαρμοστεί και Laplace smoothing, για παράδειγμα στον τύπο για τα tags:

$$P(t_n|t_{n-1}, w_{n-1}) = \frac{1 + C(t_{n-1}, w_{n-1}, t_n)}{|T| + \sum_t C(t_{n-1}, w_{n-1}, t)}$$

όπου $|T|$ το πλήθος των ετικετών.

Εφαρμογή μοντέλου

Όταν οι παράμετροι του μοντέλου έχουν μαθευτεί από τα δεδομένα εκπαίδευσης, εφαρμόζεται ο αλγόριθμος Viterbi, με τροποποίηση στον υπολογισμό των πιθανοτήτων με βάση το επαυξημένο μοντέλο, και φυσικά με την είσοδο του αλγορίθμου να μην είναι μόνο μια λίστα συμβόλων εξόδου, αλλά μια λίστα από δυάδες, μέρος του λόγου και λέξη. Η έξοδος είναι ίδιου τύπου, η λίστα με την πιθανότερη αλληλουχία κρυφών καταστάσεων, που στην ορολογία αυτής της μεθόδου καλούνται ετικέτες (tags).

2.5 Νευρωνικά Δίκτυα

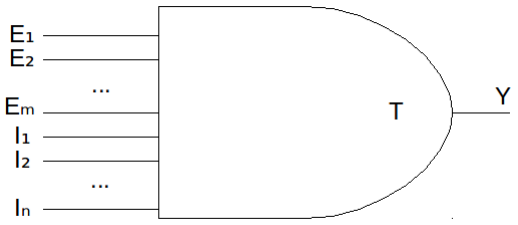
Τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks) ή απλά Νευρωνικά Δίκτυα είναι μια κατηγορία μοντέλων που χρησιμοποιούνται στη μηχανική μάθηση με επιτυχία σε πληθώρα πεδίων. Ονομάζονται έτσι γιατί είναι εμπνευσμένα από τους νευρώνες του εγκεφάλου, αλλά όσον αφορά τη μηχανική μάθηση, είναι περισσότερο ένα απλοποιημένο μοντέλο που βρίσκει εφαρμογή σε συγκεκριμένες εργασίες παρά κάτι που έχει σχέση με την πραγματική λειτουργία του ανθρώπινου εγκεφάλου, η οποία είναι περισσότερο πολύπλοκη απ᾽ ότι είναι μέχρι τώρα γνωστό στους επιστήμονες.

2.5.1 Ιστορικά στοιχεία

Η πλειοψηφία των Νευρωνικών Δικτύων που χρησιμοποιούνται σήμερα βασίζονται στο μαθηματικό μοντέλο του βιολογικού νευρώνα που επινοήθηκε το 1943 από τους McCulloch και Pitts [MP43]. Το μοντέλο του νευρώνα αυτό διέθετε m διεγερτικές εισόδους (excitory inputs) E_j, n ανασταλτικές εισόδους (inhibitory inputs) I_i και μια έξοδο Y. Εάν το άθροισμα των διεγερτικών εισόδων ξεπερνούσε ένα κατώφλι T, τότε ο νευρώνας "πυροδοτούνταν", με $Y = 1$, αλλά η παρουσία ανασταλτικής εισόδου εμπόδιζε την πυροδότηση ($Y = 0$).

$$Y = 1 \quad \text{εάν} \quad \sum_{i=1}^{n} I_i = 0 \quad \text{και} \quad \sum_{j=1}^{m} E_j \geq T$$

$$Y = 0 \quad \text{αλλιώς}$$



Σχήμα 2.4: Το μοντέλο νευρώνα των McCulloch-Pitts

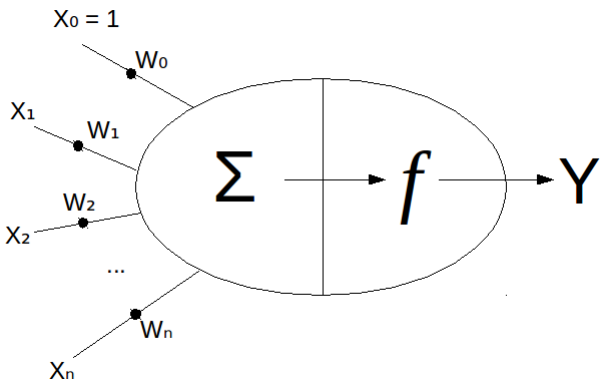
Το μοντέλο του νευρώνα τροποποιήθηκε από τον Von Neumann το 1956, ο οποίος πρότεινε οι ανασταλτικές είσοδοι να παίρνουν αρνητικές τιμές, ώστε ο νευρώνας να πυροδοτείται όταν το άθροισμα όλων των εισόδων ξεπερνούσε το κατώφλι [Pic94, p. 8]. Αργότερα, προστέθηκε στο μοντέλο η ιδέα των βαρών. Σε κάθε είσοδο αντιστοιχούσε ένας πραγματικός αριθμός w_i, με τον οποίο πολλαπλασιαζόταν η είσοδος, προτού υπεισέλθει στο άθροισμα των εισόδων. Ένα από τα πρώτα συστήματα με βάρη ήταν το ADALINE (ADAPTIVE LINEAR NEURON) [W+60].

Το 1957 ο Frank Rosenblatt εφηύρε το Perceptron [Ros58], μια ιστορική πρώτη προσέγγιση τεχνητών νευρωνικών δικτύων [Βλα+11, p. 383] και η πρώτη φορά που ένα νευρωνικό δίκτυο υλοποιήθηκε ως πρόγραμμα ηλεκτρονικού υπολογιστή (IBM 704), αντί για υλοποίηση σε ειδικά κατασκευασμένο hardware που γινόταν προηγουμένως.

2.5.2 Μοντέλο τεχνητού νευρώνα

Το μοντέλο του τεχνητού νευρώνα, στη σύγχρονη εκδοχή του, είναι βασική δομική μονάδα των Νευρωνικών Δικτύων. Μια εκδοχή αυτού του μοντέλου χαρακτηρίζεται από τα εξής:

- Δέχεται n εισόδους $x_1, x_2, \ldots, x_n$, οι οποίες παίρνουν πραγματικές τιμές.
- Δέχεται μια σταθερή είσοδο $x_0 = 1$, που καλείται πόλωση (bias).
- Διαθέτει $n+1$ βάρη $w_0, w_1, ..., w_n$, που είναι πραγματικοί αριθμοί και με τα οποία πολλαπλασιάζονται οι αντίστοιχες είσοδοι.
- Οι είσοδοι, πολλαπλασιασμένοι με τα βάρη, προστίθενται στο άθροισμα $net = \sum_{i=0}^{n} w_i x_i$.



Σχήμα 2.5: Σύγχρονο μοντέλο νευρώνα

- Στο σταθμισμένο άθροισμα των εισόδων εφαρμόζεται μια μη γραμμική συνάρτηση f, οπότε έχουμε την τελική έξοδο $y = f(net)$.
- Παραδείγματα μη γραμμικών συναρτήσεων είναι
  - Η βηματική συνάρτηση (step function)

$$f(x) = \begin{cases} 0 & \text{εάν } x \leq T \\ 1 & \text{εάν } x > T \end{cases}$$

  - Η συνάρτηση προσήμου (sign function)

$$f(x) = \begin{cases} -1 & \text{εάν } x \leq T \\ 1 & \text{εάν } x > T \end{cases}$$

  - Η σιγμοειδής συνάρτηση (sigmoid function)[2]

$$f(x) = S(x) = \frac{1}{1 + e^{-x}}$$

  - Η συνάρτηση της υπερβολικής εφαπτομένης

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

    η οποία συνδέεται με τη σιγμοειδή με τη σχέση

$$tanh(x) = 2 * S(x) - 1$$

[2]Ορθότερα, αυτή η συνάρτηση ονομάζεται λογιστική συνάρτηση (logistic function) και ανήκει στην οικογένεια των σιγμοειδών συναρτήσεων, όπου ανήκει και η υπερβολική εφαπτομένη. Όμως στην πράξη συχνά η συγκεκριμένη συνάρτηση αναφέρεται απλά ως "σιγμοειδής" και συμβολίζεται με S.

Μια εναλλακτική, και ουσιαστικά ισοδύναμη εκδοχή αυτού του μοντέλου, δε διαθέτει είσοδο πόλωσης ($x_0 = 1$), αλλά στο άθροισμα προστίθεται σταθερά b, δηλαδή έχουμε:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

Προφανώς αυτή η διατύπωση είναι ισοδύναμη, με $b = w_0$.

Ο νευρώνας αυτός μπορεί ουσιαστικά να εκφράσει μια συνάρτηση n μεταβλητών εισόδου και μιας μεταβλητής εξόδου.

Όταν στην έξοδο εφαρμόζεται συνάρτηση κατωφλίου, ώστε η έξοδος να είναι 0 ή 1, μεταξύ των συναρτήσεων που μπορούν να αναπαρασταθούν με έναν τέτοιο νευρώνα είναι οι βασικές boolean συναρτήσεις AND, OR και NOT, με κατάλληλη επιλογή των βαρών.

Αποδεικνύεται όμως ότι δεν μπορούν να αναπαρασταθούν όλες οι συναρτήσεις με ένα τέτοιο μοντέλο. Συγκεκριμένα, αναγκαία συνθήκη για να μπορεί να παρασταθεί μια συνάρτηση είναι αυτή να είναι γραμμικώς διαχωρίσιμη (linearly separable). Αυτό σημαίνει ότι πρέπει στον n-διάστατο χώρο των εισόδων, το σύνολο των εισόδων στο οποίο η έξοδος είναι 1 να μπορεί να διαχωριστεί από το σύνολο των εισόδων με έξοδο 0, με χρήση ενός υπερεπιπέδου $n - 1$ διαστάσεων. Παράδειγμα απλής συνάρτησης που δεν είναι γραμμικώς διαχωρίσιμη είναι η συνάρτηση XOR.

2.5.3 Τεχνητά Νευρωνικά Δίκτυα

Ένας νευρώνας από μόνος του είναι ένα στατιστικό μοντέλο που μπορεί να εκπαιδευτεί. Μια οργανωμένη δομή από περισσότερους του ενός νευρώνες αποτελεί ένα Τεχνητό Νευρωνικό Δίκτυο. Οι νευρώνες σε αυτά είναι συνήθως οργανωμένοι σε επίπεδα (layers).

Για παράδειγμα, ένα δίκτυο που αποτελείται από m νευρώνες, με όλους συνδεδεμένους στις ίδιες n εισόδους, είναι ένα νευρωνικό δίκτυο δύο επιπέδων. Οι n είσοδοι ονομάζονται επίπεδο εισόδων (input layer) και το επίπεδο των m νευρώνων αποτελεί το επίπεδο εξόδου (output layer). Κάποιες φορές αυτό το δίκτυο καλείται και perceptron ενός επιπέδου (single-layer perceptron), γιατί διαθέτει νευρώνες μόνο σε ένα επίπεδο (αυτό της εξόδου).

Ένα τέτοιο δίκτυο με μόνο ένα επίπεδο από νευρώνες περιορίζεται όπως και ο μεμονωμένος νευρώνας στο να μπορεί να αναπαραστήσει μόνο γραμμικώς διαχωρίσιμες συναρτήσεις.

Ο περιορισμός αυτός υπερβαίνεται εάν χρησιμοποιηθεί ένα ενδιάμεσο επίπεδο από νευρώνες. Οι είσοδοι αυτών θα είναι συνδεδεμένοι με τις n εισόδους του συστήματος, ενώ οι έξοδοί τους θα είναι συνδεδεμένοι με τις εισόδους των νευρώνων του επόμενου επιπέδου. Αυτό το νέο επίπεδο καλείται κρυφό επίπεδο (hidden layer), και το νευρωνικό δίκτυο καλείται δίκτυο τριών επιπέδων.

2.5.4 Εκπαίδευση Νευρωνικών Δικτύων

Ο στόχος της εκπαίδευσης ενός νευρωνικού δικτύου είναι να επιλεγούν τιμές για τα βάρη του ώστε να παράγονται οι σωστές έξοδοι για κάθε είσοδο. Για να εκπαιδευτεί ένα νευρωνικό δίκτυο, απαιτείται ένα σύνολο παραδειγμάτων. Έστω τα παραδείγματα $E_1, E_2, \ldots, E_k, \ldots$. Κάθε παράδειγμα E_k αποτελείται από το διάνυσμα εισόδων του παραδείγματος $X_k = [x_{k1}x_{k2}\ldots x_{kn}]$ και το διάνυσμα των σωστών εξόδων του δικτύου $T_k = [t_{k1}t_{k2}\ldots t_{km}]$ για τη δεδομένη είσοδο.

Ο πιο βασικός αλγόριθμος εκπαίδευσης των Νευρωνικών Δικτύων ονομάζεται ανάστροφη μετάδοση λάθους (back propagation). Αυτός λειτουργεί σε δύο φάσεις:

1. Φάση προσθίου περάσματος (forward pass): Η είσοδος X_k εφαρμόζεται στους νευρώνες του πρώτου επιπέδου, και παράγονται οι ενδιάμεσες έξοδοι. Οι έξοδοι κάθε επιπέδου διαδίδονται στις αντίστοιχες εισόδους του επόμενου επιπέδου, μέχρι να παραχθεί ένα διάνυσμα εξόδου $Y_k = [y_{k1}y_{k2}\ldots y_{km}]$.

2. Φάση ανάστροφου περάσματος (backward pass) ή ανάστροφης μετάδοσης (back propagation): Αφού έχουμε την έξοδο που δίνει το δίκτυο αλλά και τη σωστή έξοδο, μπορούμε να υπολογίσουμε ένα μέτρο του σφάλματος που κάνει το δίκτυο αυτή τη στιγμή. Αυτό γίνεται με το τετραγωνικό σφάλμα μεταξύ της εξόδου Y_k και της σωστής τιμής T_k, που δίνεται από τον τύπο $E_k = \sum_{j=1}^{m}(t_{km} - y_{km})^2$. Αυτό το σφάλμα είναι συνάρτηση της εισόδου, της σωστής εξόδου και των βαρών σε κάθε επίπεδο του δικτύου. Υπολογίζοντας τις μερικές παραγώγους του τετραγωνικού σφάλματος ως προς κάθε βάρος του δικτύου, μπορεί να εφαρμοστεί η μέθοδος βελτιστοποίησης επικλινούς καθόδου (gradient descent optimization), ούτως ώστε το κάθε βάρος να διορθώνεται με βάση την ποσότητα που του αναλογεί. Στα βάρη του δικτύου προστίθενται οι διορθώσεις τους και ο αλγόριθμος συνεχίζει ξανά με την πρώτη φάση.

Τα κριτήρια σταματήματος του αλγορίθμου μπορούν να οριστούν με διάφορους τρόπους: Ελαχιστοποίηση του τετραγωνικού σφάλματος κάτω από προκαθορισμένο όριο, ελαχιστοποίηση της βελτίωσης του σφάλματος ή απλά μέγιστος αριθμός επαναλήψεων. Σημειώνεται ότι τα βάρη είναι δυνατό να διορθώνονται μετά από κάθε παράδειγμα, η να διορθώνονται χρησιμοποιώντας το μέσο τετραγωνικό σφάλμα μόνο αφού τα σφάλματα για όλα τα παραδείγματα έχουν υπολογιστεί. Η πρώτη μέθοδος προσφέρει μεγαλύτερη ταχύτητα εκπαίδευσης, αλλά λιγότερη σταθερότητα σε σχέση με τη δεύτερη. Κάθε πέρασμα εκπαίδευσης που κάνει το μοντέλο από όλα τα παραδείγματα, καλείται εποχή (epoch).

2.5.5 Χρήση δικτύου τριών επιπέδων στην κατηγοριοποίηση κειμένων

Το μοντέλο του Νευρωνικού Δικτύου τριών επιπέδων μπορεί να χρησιμοποιηθεί για την κατηγοριοποίηση κειμένων, για την υλοποίηση αλγορίθμου επιτηρούμενης μάθησης. Αφού τα δίκτυα αυτά μαθαίνουν συναρτήσεις που αντιστοιχίζουν διανύσματα εισόδων πραγματικών αριθμών σε διανύσματα εξόδου, πρέπει απλά να περιγράψουμε πως αναπαρίσταται το κείμενο σαν διάνυσμα σταθερής διάστασης, και πως η κατηγοριοποίηση περιγράφεται στο διάνυσμα εξόδου.

Αρχίζοντας από την περίπτωση της εξόδου, το δίκτυό μας θα έχει ως έξοδο ένα διάνυσμα δύο διαστάσεων, το $Y = [y_1 y_2]$. Η πρώτη συνιστώσα y_1 θα έχει τιμή 1 για τη θετική γνώμη, και 0 για την αρνητική. Η δεύτερη συνιστώσα το αντίθετο. Αυτή η επιλογή φαίνεται και στον παρακάτω πίνακα:

Άποψη κειμένου	Σωστή έξοδος
Θετική	[10]
Αρνητική	[01]

Για την αναπαράσταση του κειμένου εισόδου, η πιο απλή περίπτωση είναι η αναπαράσταση ως bag-of-words. Το διάνυσμα X έχει μήκος όσο το πλήθος των λέξεων που θεωρούμε ότι ανήκουν στο λεξιλόγιο. Εάν μια λέξη εμφανίζεται στο κείμενο εισόδου, τότε η συνιστώσα του διανύσματος που της αντιστοιχεί παίρνει την τιμή 1, αλλιώς την τιμή 0.

Το πρόβλημα εδώ σε σχέση με π.χ. τον αλγόριθμο Naive Bayes είναι ότι το μήκος του διανύσματος X πρέπει να έχει καθοριστεί προτού καν εκπαιδευτεί το δίκτυο με το πρώτο παράδειγμα. Έτσι, χρειάζεται να γίνει ένα πρώτο πέρασμα στα κείμενα του συνόλου εκπαίδευσης ώστε να μαθευτεί το λεξιλόγιο, και να αποδοθεί μια συγκεκριμένη θέση στο διάνυσμα X σε κάθε λέξη. Επίσης, επειδή το μέγεθος του διανύσματος εισόδου έχει μεγάλη επίδραση στο χρόνο εκπαίδευσης, αναγκαστικά το μέγεθος του λεξιλογίου πρέπει να περιορίζεται, αν δεν είναι επιθυμητή η πολύ αργή εκπαίδευση του δικτύου.

Κεφάλαιο 3

Υλοποίηση

Η υλοποίηση των πειραμάτων έγινε στη γλώσσα προγραμματισμού Python, λόγω της ευκολίας χρήσης και της ταχύτητας ανάπτυξης πρωτοτύπων. Μειονέκτημα αποτελεί η ταχύτητα εκτέλεσης των προγραμμάτων, όντας interpreted γλώσσα. Εντούτοις, μπορούν να χρησιμοποιηθούν ειδικευμένες βιβλιοθήκες για αριθμητικούς υπολογισμούς γραμμικής άλγεβρας (π.χ. πολλαπλασιασμός πινάκων) αλλά και συμβολικών υπολογισμών (π.χ. διαφόριση συναρτήσεων ως προς κάποια παράμετρο).

3.1 Υλοποίηση Naive Bayes

Ο αλγόριθμος Naive Bayesian είναι αρκετά απλός και μπορεί να υλοποιηθεί σε ένα μόνο αρχείο κώδικα. Είναι ενδιαφέρον όμως το γεγονός ότι παρά την απλότητά του, δίνει καλά αποτελέσματα και έχει πλεονεκτήματα, όπως η γρήγορη μάθηση αλλά και η δυνατότητα να επισκοπηθούν οι παράμετροι που έχει μάθει.

Κάποιες λεπτομέρειες που αξίζει να προσεχθούν κατά την υλοποίηση είναι:

- Το πως ακριβώς χωρίζονται τα κείμενα σε λέξεις. Η πιο απλή μέθοδος είναι να θεωρείται ξεχωριστή λέξη ό,τι βρίσκεται ανάμεσα σε whitespace. Μετά από αυτό το χωρισμό όμως είναι καλύτερα να χωρίζονται και τα σημεία στίξης που ακολουθούν λέξεις χωρίς κενά διαστήματα, και να αποτελούν μια ξεχωριστή λέξη.

- Το αν θα γίνεται κανονικοποίηση ως προς μικρά-κεφαλαία, και, για την ελληνική γλώσσα, αφαίρεση τόνων και κανονικοποίηση τελικού σίγμα. Η κανονικοποίηση ενισχύει τη γνώση του μοντέλου για τα διάφορα features, αφού λέξεις που εμφανίζονται διαφορετικά αλλά είναι ουσιαστικά ίδιες λογίζονται όλες στο ίδιο feature, για το οποίο η γνώση μας είναι πλέον καλύτερη, έχοντας περισσότερα δεδομένα. Απ' την άλλη μπορεί διαφορετικές εκδοχές της ίδιας λέξης να αξίζει να έχουν διαφορετικά βάρη π.χ.

η λέξη "τέλεια" με την λέξη "ΤΕΛΕΙΑ", όπου ίσως η δεύτερη να δείχνει μεγαλύτερη ένταση στη θετική γνώμη.

Θεωρούμε πάντως πως η κανονικοποίηση αυτή αξίζει να γίνεται, ειδικά καθώς οι χρήστες του διαδικτύου γράφουν με μια ποικιλία από στυλ, για παράδειγμα χωρίς τόνους ή όλα κεφαλαία.

- Το αν θα αποκλείονται features τα οποία έχουν συναντηθεί πολύ λίγες φορές.
- Το αν θα εφαρμοστεί add-one smoothing, ώστε να μετριάζεται η επιρροή features με λίγες εμφανίσεις.
- Το αν τα features θα είναι σκέτες λέξεις, bigrams, ή n-grams με $n > 2$. Επίσης είναι δυνατό να χρησιμοποιούνται στο ίδιο μοντέλο, ταυτόχρονα, και μεμονωμένες λέξεις και bigrams, μια προσέγγιση που όπως θα δούμε στα πειραματικά αποτελέσματα στην ενότητα 4.3 δίνει τα καλύτερα αποτελέσματα.

3.2 Υλοποίηση Hidden Markov Model

Το απλό Hidden Markov Model, καθώς και το Lexicalized Hidden Markov Model Integrating Part-of-Speech, υλοποιήθηκαν στη γλώσσα προγραμματισμού Python. Για το καθένα γράφτηκε μια κλάση Python, με ονόματα HMM και LexicalizedHMM αντίστοιχα, στα αρχεία hmm.py και lexicalizedHMM.py αντίστοιχα. Το interface των δύο αυτών κλάσεων είναι παρόμοιο και είναι ευκολότερο να παρουσιαστεί με ένα απλοϊκό παράδειγμα χρήσης. Για την περίπτωση του απλού HMM:

Listing 3.1: Παράδειγμα χρήσης κλάσης HMM

```
from hmm import HMM
import math

m = HMM()   # Αρχικοποίηση

# Τροφοδότηση με δεδομένα
m.addToModel(
   [('BG','I'), ('BG','am'), ('POSITIVE','good')]
)

# Η εκπαίδευση συμβαίνει αθροιστικά
m.addToModel(
   [('BG','You'), ('BG','are'), ('POSITIVE','great')]
)
```

```
# Εκτύπωση στοιχείων μοντέλου
m.debug()


# Εφαρμογή σε αταξινόμητα δεδομένα
seq, pr = m.viterbi(['You', 'are', 'good'])

# Πιθανότερη αλληλουχία καταστάσεων:
print 'I guess:', seq
# Τυπώνει ['BG', 'BG', 'POSITIVE']

# Λογαριθμική πιθανότητα της
# παραπάνω αλληλουχίας:
print 'With probability (log):', pr

# Σε γραμμική κλίμακα:
print 'Non-log:', math.exp(pr)
```

Η κλάση LexicalizedHMM χρησιμοποιείται παρόμοια, όμως η μορφή των δεδομένων εισόδου είναι διαφορετική. Τόσο τα δεδομένα εκπαίδευσης, όσο και τα δεδομένα εφαρμογής πρέπει να συνοδεύονται από το μέρος του λόγου. Αυτή η εργασία, όπως βέβαια και ο χωρισμός του κειμένου σε λέξεις, πρέπει να γίνει από άλλο κομμάτι του κώδικα. Ένα αντίστοιχο τετριμμένο παράδειγμα χρήσης, για να φανεί το interface της κλάσης:

Listing 3.2: Παράδειγμα χρήσης κλάσης LexicalizedHMM

```
from lexicalizedHMM import LexicalizedHMM

m = LexicalizedHMM()

m.addToModel([
  ('BG', 'PROVERB', 'You'),
  ('BG', 'VERB', 'are'),
  ('POSITIVE', 'ADJECTIVE', 'good')
])

m.addToModel([
  ('BG', 'PROVERB', 'I'),
  ('POSITIVE', 'VERB', 'like'),
  ('BG', 'NOUN', 'cars')
])

seq, pr = m.viterbi([
```

```
    ('PROVERB', 'You'),
    ('VERB', 'are'),
    ('ADJECTIVE', 'great'),
])

print 'I guess:', seq
# Τυπώνει ['BG', 'BG', 'POSITIVE']

print 'With probability (log):', pr
```

Στο παραπάνω παράδειγμα το σύνολο ετικετών μέρους του λόγου είναι πολύ απλό (PROVERB, VERB, ADJECTIVE, NOUN). Το μοντέλο δουλεύει με οποιοδήποτε σύνολο μερών του λόγου, που παρέχεται από άλλο σύστημα, αρκεί βέβαια το ίδιο σύνολο ετικετών να χρησιμοποιηθεί στην εκπαίδευση και στη χρήση του μοντέλου. Είναι προφανές ότι το μοντέλο δεν έχει το ίδιο γνώση του τι είναι το μέρος του λόγου (ή επίσης του τι είναι η θετική-αρνητική γνώμη). Επίσης, οι λέξεις μπορούν να είναι σε οποιαδήποτε γλώσσα, και μπορούν να είναι είτε τύπου str ή τύπου unicode (όλος ο κώδικας της εργασίας γράφτηκε στην έκδοση 2 της Python· στην Python 3 χρειάζεται μόνο ο τύπος str, που μπορεί να αποθηκεύσει σωστά συμβολοσειρές από χαρακτήρες Unicode).

Στις επόμενες υποενότητες θα περιγράψουμε αναλυτικότερα τη λειτουργία των κλάσεων αυτών, περιοριζόμενοι στην κλάση LexicalizedHMM, αφού η κλάση HMM δεν είναι παρά μια απλούστερη εκδοχή της.

3.2.1 Ορισμός και Αρχικοποίηση

Όπως αναφέρθηκε στην υποενότητα 2.4.7, πραγματοποιείται γραμμική εξομάλυνση (interpolation) μεταξύ των πιθανοτήτων που ενσωματώνουν την προηγούμενη λέξη, και των πιο γενικών που δεν εξαρτώνται από αυτή. Οι συντελεστές λ, α και β λαμβάνουν τις τιμές τους αντίστοιχα μέσω των παραμέτρων tag_coeff pos_coeff και word_coeff του κατασκευαστή της κλάσης LexicalizedHMM, με προεπιλεγμένη τιμή για όλους το 0.5.

Η αρχικοποίηση περιλαμβάνει τον ορισμό λεξικών όπου μετρώνται οι συχνότητες των διαφόρων μεταβάσεων και αρχικών εμφανίσεων ετικετών, μερών του λόγου και λέξεων, όπως απαιτείται για να υπολογιστούν οι πιθανότητες που αναφέρθηκαν στο κεφάλαιο της θεωρίας (υποενότητα 2.4.7). Στο παρακάτω τμήμα κώδικα επισημαίνεται ο ρόλος των λεξικών/μετρητών σε σχόλια, ενώ με μία ανάγνωση του τμήματος κώδικα της εκπαίδευσης (Listing 3.4) μπορεί κανείς να καταλάβει και τον τρόπο χρήσης τους. defaultdict(**int**) που σημαίνει πως όποτε γίνεται προσπέλαση ενός ανύπαρκτου κλειδιού, αντί να εγερθεί το Exception KeyError, δημιουργείται και επιστρέφεται μια τιμή με τον κατασκευαστή **int**(), δηλαδή η τιμή 0. Αυτό το χαρακτηριστικό βολεύει πολύ ώστε στις επόμενες μεθόδους να αποφεύγεται επαναλαμβανόμενος κώδικας χειρισμού των ανύπαρκτων κλειδιών.

Listing 3.3: Ορισμός κλάσης LexicalizedHMM

```
from collections import defaultdict

class LexicalizedHMM(object):
    def __init__(self, word_coeff=0.5,
                 pos_coeff=0.5, tag_coeff=0.5):
        # Interpolation coefficient between lexicalized and
        # non-lexicalized models.
        self.word_coeff = word_coeff
        self.pos_coeff = pos_coeff
        self.tag_coeff = tag_coeff

        # Numerator of P(w_i|t_i, s_i, w_{i-1})
        # i.e. C(w_i, t_i, s_i, w_{i-1})
        # dictionary (str, str, str, str) -> int
        # (w_i, t_i, s_i, w_{i-1}) -> Count of occurrences
        self.CW = defaultdict(int)

        # Numerator of P(s_i|t_i, w_{i-1})
        # i.e. C(s_i, t_i, w_{i-1})
        # dictionary (str, str, str) -> int
        # (s_i, t_i, w_{i-1}) -> Count of occurrences
        self.CS = defaultdict(int)

        # Numerator of P(t_i|t_{i-1}, w_{i-1})
        # i.e. C(t_i, t_{i-1}, w_{i-1})
        # dictionary (str, str, str) -> int
        # (t_i, t_{i-1}, w_{i-1}) -> Count of occurrences
        self.CT = defaultdict(int)


        # Denomitator counts

        # self.DW is same as self.CS
        self.DS = defaultdict(int)
        self.DT = defaultdict(int)


        # Initial probabilities for T
        # str -> int
        # t_1 -> Count of occurrences
        self.InitT = defaultdict(int)
```

```
# Appearances of POS when there's no previous token.
# So POS depends only on Tag.
# (str, str) -> int
# (t_1, s_1) -> Count of occurrences
self.InitS = defaultdict(int)

# Appearances of word when there's no previous token.
# So Word depends only on Tag and POS.
# (str, str, str) -> int
# (w_1, t_1, s_1) -> Count of occurrences
self.InitW = defaultdict(int)

# Denominators of initial probabilities.
self.DInitT = 0 # this is just a counter
# self.DInitS is same as self.InitT
# self.DInitW is same as self.InitS

# Lower-order (non-lexicalized) models:

# Numerator of P(w_i|t_i, s_i)
self.LOW = defaultdict(int)
# Numerator of P(s_i|t_i)
self.LOS = defaultdict(int)
# Numerator of P(t_i|t_{i-1})
self.LOT = defaultdict(int)

# Denominators for lower order models:
self.DLOW = self.LOS
self.DLOS = defaultdict(int)
self.DLOT = defaultdict(int)

# Set of all tags:
self.alltags = set()
# Set of all parts of speech:
self.allpos = set()
# Set of all words:
self.allwords = set()

# This allows to set restrictions and make some transitions
# completely impossible (if the model requires this).
self.bannedTransitions = set()
```

3.2.2 Εκπαίδευση

Η μέθοδος addToModel(sequence) εκπαιδεύει το μοντέλο με βάση ταξινομημένα δεδομένα. Για την κλάση LexicalizedHMM, το όρισμα sequence είναι λίστα από τριάδες της μορφής (tag, part_of_speech, word), δηλαδή ένα κείμενο που έχει διαχωριστεί σε λέξεις, και σε κάθε λέξη έχει αποδοθεί η κατάλληλη ετικέτα καθώς και το μέρος του λόγου.

Η κυρίως εργασία αυτής της μεθόδου είναι αρκετά απλή, αφού μετρά τις συχνότητες των διάφορων συνδυασμών μεταβάσεων, καθώς και των αρχικών εμφανίσεων ετικετών, μερών του λόγου και λέξεων.

Listing 3.4: Εκπαίδευση μοντέλου LexicalizedHMM

```
def addToModel(self, sequence):
    # Update model with training data
    #
    # Argument sequence is comprised of tuples
    # in the form (tag, part-of-speech, word).
    # These sequences represent text that has
    # already been tokenized and labeled.

    sequence = list(sequence)
    if not sequence:
        return

    t, s, w = sequence[0]

    self.InitT[t] += 1
    self.InitS[(s, t)] += 1
    self.InitW[(w, s, t)] += 1

    # Denominators for initial probabilities:
    self.DInitT += 1

    it = iter(sequence)
    it.next()

    self.alltags.add(t)
    self.allpos.add(s)
    self.allwords.add(w)

    tprev, sprev, wprev = t, s, w
    for (t, s, w) in it:
        # The counts for the nominators
        # of the probabilities:
```

```
        self.CW[(w, t, s, wprev)] += 1
        self.CS[(s, t, wprev)] += 1
        self.CT[(t, tprev, wprev)] += 1

        # The counts for the denominators
        # of the probabilities:

        # self.DW is same as self.CS
        self.DW = self.CS
        self.DS[(t, wprev)] += 1
        self.DT[(tprev, wprev)] += 1


        # Lower-order (non-lexicalized) models
        # (i.e. without taking wprev into account).
        self.LOW[(w, t, s)] += 1
        self.LOS[(s, t)] += 1
        self.LOT[(t, tprev)] += 1

        # Denominators for lower order models.
        # self.DLOW is same as self.LOS
        self.DLOW = self.LOS
        self.DLOS[t] += 1
        self.DLOT[tprev] += 1
        # The above two dictionaries are ALMOST the same.

        self.alltags.add(t)
        self.allpos.add(s)
        self.allwords.add(w)

        tprev, sprev, wprev = t, s, w
```

3.2.3 Εφαρμογή

Η εφαρμογή του εκπαιδευμένου μοντέλου γίνεται από τη μέθοδο viterbi. Για να δείξουμε πως υλοποιήθηκε θα παραθέσουμε δύο προκαταρκτικά κομμάτια κώδικα πρώτα. Τρεις απλές βοηθητικές συναρτήσεις, που υλοποιούν το Laplace smoothing και το linear interpolation. Επίσης, υπενθυμίζουμε πως ό,τι εμφανίζεται σαν πολλαπλασιασμός πιθανοτήτων στον αλγόριθμο 1 υλοποιείται σαν πρόσθεση των λογαρίθμων των πιθανοτήτων, ώστε να μην έχουμε underflow στους floating-point αριθμούς της Python, γι αυτό και οι βοηθητικές αυτές συναρτήσεις επιστρέφουν λογαρίθμους.

Listing 3.5: Βοηθητικές συναρτήσεις

```
import math
log = math.log

# nums: number of possible values for numerator
def laplace(numerator, denominator, nums):
    return (numerator + 1.0) / (denominator + nums)
def logLaplace(n, d, ns):
    return log(laplace(n, d, ns))
def logInterp(t, term1, term2):
    return log(t*term1 + (1-t)*term2)
```

Στη συνέχεια, οι παρακάτω μέθοδοι της κλάσης LexicalizedHMM υπολογίζουν τις πιθανότητες που εμφανίζονται στον αλγόριθμο Viterbi, εφαρμόζοντας τη γραμμική εξομάλυνση και το Laplace smoothing στις κατάλληλες συχνότητες μεταβάσεων και εμφανίσεων των ετικετών, μερών του λόγου και λέξεων, όπως αυτές μετρήθηκαν κατά την εκπαίδευση. Οι υπογραφές των μεθόδων initPW, initPS, initPT, probW, probS και probT είναι γραμμένες έτσι ώστε να αντιστοιχούν στην πιθανότητα εμφάνισης του πρώτου ορίσματος δεδομένων των άλλων, δηλαδή όταν γράφουμε self.probT(t, tprev, wprev) αυτό σημαίνει $P(t_i|t_{i-1}, w_{i-1})$.

Listing 3.6: Μέθοδοι υπολογισμού πιθανοτήτων

```
def initPW(self, w, s, t):
  return logLaplace(self.InitW[w,s,t], self.InitS[s,t], len(self.allwords))

def initPS(self, s, t):
  return logLaplace(self.InitPS[s,t], self.InitT[t], len(self.allpos))

def initPT(self, t):
  return logLaplace(self.InitPT[t], self.DInitT, len(self.alltags))

def probW(self, w, t, s, wprev):
  return logInterp(
    self.word_coeff,
    laplace(self.PW[w,t,s,wprev], self.CS[s,t,wprev], len(self.allwords)),
    laplace(self.LOW[w,t,s], self.DLOW[t,s], len(self.allwords)))

def probS(self, s, t, wprev):
  return logInterp(
    self.pos_coeff,
    laplace(self.PS[s,t,wprev], self.DS[t,wprev], len(self.allpos)),
    laplace(self.LOS[s,t], self.DLOS[t], len(self.allpos)))

def probT(self, t, tprev, wprev):
  if (tprev, t) in self.bannedTransitions:
    return float('-inf')

  return logInterp(
    self.tag_coeff,
    laplace(self.PT[t,tprev,wprev], self.DT[tprev,wprev], len(self.alltags)),
    laplace(self.LOT[t,tprev], self.DLOT[tprev], len(self.alltags)))
```

Αφού προετοιμάσαμε το έδαφος μέσω των βοηθητικών συναρτήσεων, υλοποιούμε τον αλγόριθμο Viterbi (Αλγόριθμος 1) για την εύρεση της πιθανότερης ακολουθίας καταστάσεων. Παρατηρούμε ότι αντί για πολλαπλασιασμό έχουμε πρόσθεση λογαρίθμων και ότι οι υπολογισμοί των πιθανοτήτων έχουν περισσότερες συνιστώσες λόγω του επαυξημένου μοντέλου μεταβάσεων.

Listing 3.7: Υλοποίηση εύρεσης πιθανότερης ακολουθίας καταστάσεων στο LexicalizedHMM

```
def viterbi(self, ):
# M.viterbi(tokens) -> (prob, sequence)
#
# tokens is a list in the form (part-of-speech, word).
# Returns the most probable sequence of hidden
# states (tags) as a list and the corresponding
# probability (its logarithm).

  V1 = dict()
  W1 = dict()
  V2 = dict()
  W2 = dict()

  it = iter(tokens)
  s, w = it.next()

  for t in self.alltags:
    V1[t] = self.initPT(t) + self.initPS(s,t) + self.initPW(w,s,t)
    W1[t] = []

  sprev, wprev = s, w

  for i in range(1,len(tokens)):
    s, w = it.next()
    for t in self.alltags:
      pr,choice = max((self.probT(t,x,wprev) + V1[x], x) for x in self.alltags)

      W2[t] = W1[choice] + [choice]
      V2[t] = self.probS(s,t,wprev) + self.probW(w,t,s,wprev) + pr

    W1, W2 = W2, W1
    V1, V2 = V2, V1
    sprev, wprev = s, w

  best_pr, best_last_tag = max((pr,t) for t,pr in V1.items())
  best_sequence = W1[best_last_tag] + [best_last_tag]

  return best_sequence, best_pr
```

3.3 Υλοποίηση Νευρωνικών Δικτύων

3.3.1 Επιλογή εργαλείων

Για την εκτέλεση πειραμάτων με χρήση νευρωνικών δικτύων, υλοποιήθηκε μια κλάση σε γλώσσα Python που μοντελοποιεί ένα νευρωνικό δίκτυο τριών επιπέδων, δηλαδή ένα δίκτυο με ένα κρυφό επίπεδο (hidden layer).

Στις πιο αρχικές υλοποιήσεις, χρησιμοποιήθηκαν οι ενσωματωμένες δομές δεδομένων της Python, κυρίως λίστες. Στη συνέχεια, για αύξηση της απόδοσης, και επειδή πρακτικά στην εκπαίδευση και εφαρμογή νευρωνικών δικτύων οι πράξεις που απαιτούνται είναι μεταξύ πινάκων (που περιέχουν τις εισόδους, τα βάρη και τις εξόδους των νευρώνων), χρησιμοποιήθηκε η βιβλιοθήκη αριθμητικών υπολογισμών NumPy [WCV11], που παρέχει αποδοτικές υλοποιήσεις για πράξεις μεταξύ πινάκων και πολλούς αριθμητικούς αλγορίθμους. Αυτή η βιβλιοθήκη μπορεί να συγκριθεί σε λειτουργικότητα με το γνωστό σύστημα MATLAB, αλλά μπορεί να χρησιμοποιηθεί από προγράμματα Python, και επίσης είναι λογισμικό ανοικτού κώδικα.

Με χρήση της NumPy, η διαφόριση της συνάρτησης τετραγωνικού σφάλματος ως προς τα βάρη του δικτύου και η εξαγωγή των τύπων για τον κανόνα δέλτα στην εκπαίδευση του δικτύου πρέπει να γίνει από τον προγραμματιστή. Μια εναλλακτική που βελτιώνει την ευκολία υλοποίησης διαφόρων αρχιτεκτονικών νευρωνικών δικτύων είναι η χρήση της βιβλιοθήκης Theano [Ber+10; Bas+12], που παρέχει τη δυνατότητα αυτόματης διαφόρισης συναρτήσεων σε συμβολικό επίπεδο. Έτσι, κανείς αρκεί να ορίσει τη δομή του νευρωνικού δικτύου ως προς την προς τα εμπρός λειτουργία του, και οι μερικές παράγωγοι του σφάλματος ως προς τα βάρη (και άλλων ενδεχομένως άλλων παραμέτρων που μαθαίνονται, όπως πίνακες συνέλιξης) υπολογίζονται αυτόματα, διευκολύνοντας την υλοποίηση της εκπαίδευσης με back propagation.

Άλλα χαρακτηριστικά της βιβλιοθήκης Theano είναι:

- Μπορεί να χρησιμοποιήσει τις δομές πινάκων της βιβλιοθήκη NumPy στους υπολογισμούς της.
- Για τις συναρτήσεις σε συμβολικό επίπεδο παράγεται κώδικας C ο οποίος μεταγλωττίζεται επί τόπου, οδηγώντας σε γρηγορότερη εκτέλεση.
- Παρέχει δυνατότητα πραγματοποίησης υπολογισμών στην GPU για ακόμα μεγαλύτερη επιτάχυνση πράξεων μεταξύ floats των 32 bit.

Στη συνέχεια θα παραθέσουμε τμήματα κώδικα που δείχνουν πως υλοποιήθηκε ένα νευρωνικό δίκτυο με αυτή τη βιβλιοθήκη σε γλώσσα Python.

3.3.2 Η κλάση του Νευρωνικού Δικτύου

Στο παρακάτω κομμάτι κώδικα (Listing 3.8) φαίνεται ο ορισμός της κλάσης NN (Neural Network) που υλοποιεί το δίκτυο τριών επιπέδων. Οι τρεις παράμετροι στην αρχικοποίηση της κλάσης είναι η διάσταση του διανύσματος εισόδου (ninput), ο αριθμός των νευρώνων στο κρυφό επίπεδο (nhidden) και η διάσταση του διανύσματος εξόδου (noutput) που ισοδυναμεί με τον αριθμό των νευρώνων στο επίπεδο εξόδου.

Τα βάρη στο κρυφό επίπεδο αναπαρίστανται με τον πίνακα w_2, διαστάσεων $ninput \times nhidden$. Η πόλωση στους $nhidden$ νευρώνες με το διάνυσμα b_2,

διάστασης $nhidden$. Τα βάρη στο επίπεδο εξόδου αποθηκεύονται στον πίνακα w_3, διαστάσεων $ninput \times nhidden$, και η αντίστοιχη πόλωση στο διάνυσμα b_3, διάστασης $noutput$.

Στο κομμάτι κώδικα παρακάτω φαίνεται η τυχαία ανάθεση αρχικών τιμών στις προαναφερθείσες παραμέτρους, με χρήση κατάλληλου εύρους τιμών [GB10]. Οι αρχικές τιμές αυτές χρησιμοποιούνται στη συνάρτηση compile_functions, η οποία κάνει την περισσότερη δουλειά του ορισμού του νευρωνικού δικτύου και φαίνεται στο επόμενο κομμάτι κώδικα (Listing 3.9).

Listing 3.8: Ορισμός και Αρχικοποίηση κλάσης NN

```
import theano
import numpy as np
import random
from math import sqrt

import theano.tensor as T

# Implementation of a standard 3-layer Neural Network using the Theano
# library.

# It learns a (W · x + b) style computation, instead of (W · x) with an
# extra x0 = +1 bias input.

class NN(object):
    """The Neural Network.
    """

    def __init__(self, ninput, nhidden, noutput):

        self.ninput  = ninput
        self.nhidden = nhidden
        self.noutput = noutput

        # Suggested ranges for weight initialization for second and
        # third layer. See:
        # http://deeplearning.net/tutorial/mlp.html
        r2 = 4.0 * sqrt(6.0 / (ninput + nhidden))
        r3 = 4.0 * sqrt(6.0 / (nhidden + noutput))

        # Uniform distribution in range (−r, r).
        def rand(r):
            return r*(-1.0 + 2*random.random())
```

```
        # Initial values of weights w and biases b for the two layers.

        self.init_w2 = np.array(
            [[rand(r2) for j in range(nhidden)]
                for i in range(ninput)])

        self.init_w3 = np.array(
            [[rand(r3) for j in range(noutput)]
                for i in range(nhidden)])

        self.init_b2 = np.array(
            [rand(r2) for i in range(nhidden)])

        self.init_b3 = np.array(
            [rand(r3) for i in range(noutput)])

        # Create the training and evaluation functions.
        self.compile_functions()
```

Στη συνάρτηση compile_functions (ολόκληρη στο Listing 3.9 παρακάτω) ορίζεται ο υπολογισμός του τελικού διανύσματος εξόδου x3 σε συμβολική μορφή, βήμα προς βήμα. Ξεκινάμε με τον ορισμό της εισόδου x1 ως διάνυσμα από doubles (T.dvector()).

```
x1 = T.dvector()
```

Επίσης ορίζουμε τα βάρη και τις πολώσεις του δικτύου w2, w3, b2, b3 ως shared variables, επειδή οι τιμές τους θα πρέπει να διατηρούνται μεταξύ των κλήσεων της συνάρτησης εκπαίδευσης (για περισσότερες πληροφορίες βλ. το Tutorial της βιβλιοθήκης Theano). Ακόμα, αυτές οι μεταβλητές αρχικοποιούνται με τις τυχαίες τιμές που επιλέχθηκαν προηγουμένως και το όρισμα name δίνει ένα όνομα που βοηθά στην αποσφαλμάτωση.

```
w2 = theano.shared(self.init_w2, name='w2')
w3 = theano.shared(self.init_w3, name='w3')

b2 = theano.shared(self.init_b2, name='b2')
b3 = theano.shared(self.init_b3, name='b3')
```

Η ενδιάμεση μεταβλητή (διάνυσμα) net2 ορίζεται ως το γινόμενο πινάκων (dot product) μεταξύ του διανύσματος εισόδου x1 και των βαρών του κρυφού επιπέδου w2, συν το διάνυσμα των πολώσεων b2. Η έξοδος του κρυφού επιπέδου x2 προκύπτει από την εφαρμογή της σιγμοειδούς συνάρτησης $\frac{1}{1+e^{-x}}$, στοιχείο-προς-στοιχείο, στο διάνυσμα net2.

```
net2 = T.dot(x1, w2) + b2
x2 = 1.0 / (1.0 + T.exp(-net2))
```

Στις επόμενες δύο γραμμές ορίζεται η σχέση της τελικής εξόδου του δικτύου x3, σε σχέση με τα βάρη w3, τις πολώσεις b3 και την είσοδο x2 στο επίπεδο εξόδου, με παρόμοιο τρόπο όπως στο κρυφό επίπεδο, δηλαδή με εφαρμογή της σιγμοειδούς συνάρτησης στο σταθμισμένο άθροισμα net3.

```
net3 = T.dot(x2, w3) + b3
x3 = 1.0 / (1.0 + T.exp(-net3))
```

Είναι σημαντικό να παρατηρήσουμε ότι μέχρι στιγμής δεν έχουν εκτελεστεί υπολογισμοί, αλλά έχει οριστεί ο υπολογισμός που θέλουμε να γίνεται σε συμβολική μορφή. Η μεταβλητή x3 δηλαδή φυλάσσει μέχρι στιγμής μόνο ένα συντακτικό δέντρο που περιγράφει πως υπολογίζεται, συναρτήσει της μοναδικής εισόδου x1, και των "μοιραζόμενων" μεταβλητών w2, w3, b2 και b3.

Η δημιουργία της συνάρτησης εφαρμογής του δικτύου στην είσοδο (self.applyfunc) συμβαίνει με την κλήση theano.function. Σε αυτή τη στιγμή, δημιουργείται και μεταγλωττίζεται κώδικας C από τη βιβλιοθήκη που αντιστοιχεί στον υπολογισμό της εξόδου x3 από την είσοδο x1, όπως ορίστηκε προηγουμένως. Το αντικείμενο (self.applyfunc) μπορεί να καλείται τώρα με ένα όρισμα, που πρέπει να αντιστοιχεί στο διάνυσμα εισόδου, και θα επιστρέφει το διάνυσμα εξόδου του δικτύου. Αυτή είναι η συνάρτηση της πρόσθιας λειτουργίας του δικτύου (feed forward), δηλαδή της εφαρμογής του.

```
self.applyfunc = theano.function(
  inputs=[x1],
  outputs=x3,
)
```

Χρειαζόμαστε να δημιουργήσουμε και τη συνάρτηση εκπαίδευσης. Αυτή έχει ως είσοδο όχι μόνο το διάνυσμα εισόδου x1, αλλά και το διάνυσμα της σωστής εξόδου. Το ορίζουμε ως y = T.dvector() δηλαδή ένα ακόμα διάνυσμα από doubles. Στη συνέχεια ορίζουμε τη συνάρτηση κόστους (cost) που η εκπαίδευση του δικτύου θα προσπαθεί να ελαχιστοποιήσει, και αυτή είναι το τετραγωνικό σφάλμα μεταξύ της εξόδου του δικτύου x3 και της σωστής εξόδου y.

```
y = T.dvector()
cost = T.sum((x3-y)**2)
```

Για να τροποποιούμε τα βάρη μετά από κάθε παράδειγμα εκπαίδευσης, χρειαζόμαστε τις μερικές παραγώγους της συνάρτησης κόστους ως προς τα βάρη. Αυτές οι παράγωγοι υπολογίζονται αυτόματα σε συμβολικό επίπεδο με τη συνάρτηση T.grad. Οι παράμετροί της είναι η συνάρτηση κόστους (cost), μια λίστα με τις μεταβλητές ως προς τις οποίες θέλουμε να υπολογιστούν οι παράγωγοι (wrt: with regards to, ως προς) και οι όροι που θεωρούνται σταθεροί (consider_constant). Δεχόμαστε τόσα αποτελέσματα όσες οι μεταβλητές ως προς τις οποίες παραγωγίσαμε. Οι τέσσερις τιμές που επιστρέφονται (grad_w2, grad_w3, grad_b2, grad_b3) είναι συναρτήσεις σε συμβολική μορφή

που για δεδομένες τιμές των x1, y, w2, w3, b2, b3 επιστρέφουν την τιμή των παραγώγων $\frac{\partial cost}{\partial w2}$, $\frac{\partial cost}{\partial w3}$, $\frac{\partial cost}{\partial b2}$ και $\frac{\partial cost}{\partial b3}$ αντίστοιχα.

```
grad_w2, grad_w3, grad_b2, grad_b3 = T.grad(
  cost=cost,
  wrt=[w2, w3, b2, b3],
  consider_constant=[x1, y]
)
```

Η συνάρτηση εκπαίδευσης δημιουργείται πάλι με την κλήση theano.function, που όμως αυτή τη φορά εκτός από εισόδους και εξόδους, έχει και τη λειτουργία της ενημέρωσης (update) των τιμών των shared μεταβλητών w2, w3, b2 και b3 κάθε φορά που καλείται. Συγκεκριμένα, οι τιμές των βαρών (και των πολώσεων) αντικαθίστανται με τις ίδιες, μειωμένες κατά τη μερική παράγωγο του κόστους που ορίστηκε προηγουμένως, πολλαπλασιασμένη κατά σταθερά που καλείται ρυθμός μάθησης (learning rate). Η επιλογή της σταθεράς αυτής παίζει πολύ σημαντικό ρόλο στην ταχύτητα με την οποία η εκπαίδευση του Νευρωνικού Δικτύου συγκλίνει σε μια σταθερή κατάσταση, αλλά και το αν αυτό θα επιτευχεθεί. Εμείς στα πειράματά μας επιλέξαμε τη σταθερή τιμή 0.5 αφού πειραματιστήκαμε και με άλλες τιμές, παρότι υπάρχουν "εξυπνότεροι" αλγόριθμοι όπου ο ρυθμός μάθησης μεταβάλλεται ανάλογα με την πορεία της εκπαίδευσης. Τελικά, το αντικείμενο self.trainfunc μπορεί να καλείται με δύο ορίσματα-διανύσματα, την είσοδο και την έξοδο για ένα παράδειγμα εκπαίδευσης.

```
learning_rate = 0.5

self.trainfunc = theano.function(
  inputs=[x1, y],
  outputs=[x3, cost],
  updates=(
    (w2, w2 - learning_rate * grad_w2),
    (w3, w3 - learning_rate * grad_w3),
    (b2, b2 - learning_rate * grad_b2),
    (b3, b3 - learning_rate * grad_b3)
  )
)
```

Παρακάτω παρατίθεται ολόκληρη η συνάρτηση:

Listing 3.9: Δημιουργία συναρτήσεων εκπαίδευσης και εφαρμογής

```
def compile_functions(self):

  x1 = T.dvector()
```

```
    w2 = theano.shared(self.init_w2, name='w2')
    w3 = theano.shared(self.init_w3, name='w3')

    b2 = theano.shared(self.init_b2, name='b2')
    b3 = theano.shared(self.init_b3, name='b3')

    net2 = T.dot(x1, w2) + b2
    x2 = 1.0 / (1.0 + T.exp(-net2))

    net3 = T.dot(x2, w3) + b3
    x3 = 1.0 / (1.0 + T.exp(-net3))

    self.applyfunc = theano.function(
      inputs=[x1],
      outputs=x3,
    )

    y = T.dvector()
    cost = T.sum((x3-y)**2)

    grad_w2, grad_w3, grad_b2, grad_b3 = T.grad(
      cost=cost,
      wrt=[w2, w3, b2, b3],
      consider_constant=[x1, y]
    )

    learning_rate = 0.5

    self.trainfunc = theano.function(
      inputs=[x1, y],
      outputs=[x3, cost],
      updates=(
        (w2, w2 - learning_rate * grad_w2),
        (w3, w3 - learning_rate * grad_w3),
        (b2, b2 - learning_rate * grad_b2),
        (b3, b3 - learning_rate * grad_b3)
      )
    )
```

Τέλος, ορίσαμε ένα απλό interface με συναρτήσεις που εκτελούν την εκπαίδευση σε ένα παράδειγμα (train), την εφαρμογή του μοντέλου (test) και μια ακόμα συνάρτηση που εφαρμόζει ένα κατώφλι ώστε το διάνυσμα εξόδου να περιέχει μόνο τις τιμές 0 ή 1 (calculate_answer).

Listing 3.10: Διεπαφή χρήσης κλάσης NN

```
def test(self, x1):
  return self.applyfunc(x1)

def train(self, x1, d):
  return self.trainfunc(x1, d)


hardlimit = np.vectorize(lambda x: 1.0 if x>0.5 else 0.0)

def calculate_answer(self, x1):
  x3 = self.test(x1)

  return self.hardlimit(x3)
```

Φυσικά για να χρησιμοποιηθεί η παραπάνω κλάση σε κείμενο πρέπει κάποιος συνδετικός κώδικας να μετατρέπει τα κείμενα εισόδου σε διανύσματα μήκους $ninput$ με βάση μια αναπαράσταση bag-of-words, όπως περιγράφηκε στην υποενότητα 2.5.5.

Εάν επίσης είναι επιθυμητό να συμπεριληφθούν και bigrams στην είσοδο, αυτά μπορούν να προστεθούν με την ίδια λογική, δηλαδή τιμή 0 εάν δεν είναι παρών στο κείμενο, 1 αν είναι.

3.4 Υλοποίηση Stemmer

Για να χρησιμοποιηθεί ως βοηθητικό εργαλείο στις προαναφερθείσες μεθόδους, αναζητήσαμε υλοποίηση stemmer για την ελληνική γλώσσα στη γλώσσα Python. Καθώς τέτοιος δεν υπήρχε, εντοπίσαμε το `mod_stemmer.php` του Κώστα Μαγαρισιώτη σε γλώσσα PHP[1], ο οποίος βασίστηκε στη δουλειά του Σπύρου Σαρούκου [Sar09], ο οποίος με τη σειρά του είχε επεκτείνει τη δουλειά του Γιώργου Νταή [Nta06]. Μεταγλωττίσαμε τον stemmer από PHP σε Python, και χρησιμοποιήσαμε αυτό το port στην εργασία μας, συγκεκριμένα στα μοντέλα Naive Bayes και Hidden Markov.

Για το stemming στην αγγλική γλώσσα, που χρειάστηκε για το σύνολο δεδομένων κριτικών ταινιών, χρησιμοποιήσαμε την υλοποίηση του Porter stemmer [Por80] που περιέχεται στη βιβλιοθήκη NLTK [BKL09].

[1] `http://www.magarisiotis.gr/index.php/portfolio`

3.5 Υλοποίηση Part-of-Speech Tagger

Επίσης ως βοηθητικό εργαλείο, απαραίτητο για το μοντέλο Lexicalized HMM Integrating POS, χρειαστήκαμε μια υλοποίηση επισημειωτή μέρους του λόγου για την ελληνική γλώσσα. Χρησιμοποιήσαμε δύο μοντέλα για τη δουλειά αυτή. Το πρώτο το εκπαιδεύσαμε μόνοι μας, επιλέγοντας το σύνολο των μερών του λόγου (πίνακας 3.1), δημιουργώντας ένα μικρό σύνολο εκπαίδευσης από κείμενα όπου κάθε λέξη σημειώθηκε με το μέρος του λόγου της, και εκπαιδεύοντας ένα μοντέλο τύπου Averaged Perceptron Tagger. Η υλοποίηση Averaged Perceptron που χρησιμοποιήθηκε ήταν του Matthew Honnibal και συμπεριλαμβάνεται στη βιβλιοθήκη `textblob` που είναι γραμμένη σε Python.

Το σύνολο εκπαίδευσης που δημιουργήσαμε αποτελούνταν από 238 αρχεία μέσα από το σύνολο κριτικών ξενοδοχείων που είχαμε συλλέξει (βλέπε 4.1.1). Σε αυτά σημειώθηκε το μέρος του λόγου για 4909 λέξεις, με τη συχνότητα των μερών του λόγου να φαίνεται στον πίνακα 3.2.

Ετικέτα μέρους του λόγου	Μέρος του λόγου των ελληνικών που αντιπροσωπεύει
ARTICLE	Άρθρο
NOUN	Ουσιαστικό
VERB	Ρήμα
ADJECTIVE	Επίθετο
ADVERB	Επίρρημα
PARTICIPLE	Μετοχή
PREPOSITION	Πρόθεση
PRONOUN	Αντωνυμία
CONJUNCTION	Σύνδεσμος
PARTICLE	Μόριο
PUNCTUATION	Σημείο στίξης
OTHER	Άλλο

Πίνακας 3.1: Επιλεγμένο set μερών του λόγου για την ελληνική γλώσσα

Αυτό το μοντέλο δε δούλευε τέλεια, αλλά η ορθότητά του ήταν αρκετή ώστε να βοηθήσει στην απόδοση του μοντέλου HMM όταν αυτό χρειαζόταν να ξέρει το μέρος του λόγου των λέξεων.

Το δεύτερο, πολύ πληρέστερο μοντέλο που χρησιμοποιήσαμε ήταν το σύστημα part-of-speech tagging που παρέχεται ελεύθερα για ερευνητικούς σκοπούς ως διαδικτυακή υπηρεσία από το Ινστιτούτο Επεξεργασίας του Λόγου. Αυτό το μοντέλο διέθετε αρκετά μεγαλύτερο σύνολο ετικετών, που προσδιόριζαν όχι μόνο το μέρος του λόγου βάσει των 10 μερών του λόγου, αλλά και βάσει άλλων γραμματικών στοιχείων της λέξης όπως π.χ. γένος, πτώση, αριθμός κ.λ.π. Επίσης, είχε εκπαιδευτεί σε πολύ μεγαλύτερο σύνολο δεδομένων

Ετικέτα μέρους του λόγου	Αριθμός επισημάνσεων
NOUN	1183
PUNCTUATION	691
ARTICLE	636
ADJECTIVE	570
VERB	456
CONJUNCTION	376
ADVERB	341
PREPOSITION	198
PRONOUN	176
OTHER	136
PARTICLE	108
PARTICIPLE	38

Πίνακας 3.2: Στατιστικά συνόλου εκπαίδευσης απλού part-of-speech tagger

και ήταν γενικά πιο αξιόπιστο. Για να το χρησιμοποιήσουμε, χρειαστήκαμε τη βιβλιοθήκη suds της Python που υλοποιεί το πρωτόκολλο SOAP, το οποίο χρησιμοποιούσε η διαδικτυακή υπηρεσία.

Για να μην επικοινωνούμε συνέχεια με τον server σε κάθε πείραμα που κάναμε, χρησιμοποιήσαμε τη δυνατότητα batch επεξεργασίας αρχείων που παρέχει το Web Interface του συστήματος (`http://nlp.ilsp.gr/soaplab2-axis/`), ώστε να αποθηκεύσουμε τοπικά τα αποτελέσματα του part-of-speech tagging για κάθε ένα από τα κείμενα όπου αυτό θα χρειαζόταν. Αυτά δεν ήταν όλο το σύνολο δεδομένων κριτικών ξενοδοχείων, αλλά το υποσύνολο που συμμετείχε στα πειράματα του μοντέλου Lexicalized HMM Integrating Part-of-Speech, που αριθμούσε μερικές εκατοντάδες μόνο κείμενα, γιατί απαιτήθηκε η δικιά μας χειρωνακτική απόδοση ετικετών (βλέπε και 4.4.2).

Κεφάλαιο 4

Πειραματικά αποτελέσματα

4.1 Σύνολα δεδομένων

Ένα ζητούμενο για την εφαρμογή μηχανικής μάθησης στην ανάλυση άποψης είναι η ύπαρξη ταξινομημένων συνόλων δεδομένων στα οποία μπορούν να δοκιμάζονται οι διάφοροι αλγόριθμοι και μέθοδοι. Η απλούστερη μέθοδος, δηλαδή η χειρωνακτική εύρεση και κατηγοριοποίηση σχολίων από τους ερευνητές, είναι ιδιαίτερα χρονοβόρα. Πιθανολογούμε ότι μια μικρή ομάδα ερευνητών (1-3), μπορεί να παράξει κατηγοριοποιημένο σύνολο δεδομένων της τάξης μεγέθους των εκατοντάδων κειμένων (100-1000) καταβάλλοντας συνολική προσπάθεια της τάξης των δεκάδων ωρών (10-100). Οι χρονικές απαιτήσεις επίσης μεγαλώνουν σημαντικά εάν απαιτούνται λεπτομερείς σημειώσεις πάνω στο κείμενο (π.χ. επισήμανση συγκεκριμένων λέξεων-φράσεων και απόδοσής τους σημασιολογικών κατηγοριών).

Μια μέθοδος για αποδοτική συλλογή μεγάλων κατηγοριοποιημένων συνόλων είναι η συλλογή τους από ιστοσελίδες με κριτικές, όπου το κείμενο της κριτικής συνοδεύεται από κάποιου είδους βαθμολογίας ή χαρακτηρισμού, ώστε να μπορούν να αποδοθούν κατηγορίες θετικής ή αρνητικής γνώμης στα κείμενα. Έτσι, για την παρούσα διπλωματική συγκεντρώσαμε ένα σύνολο δεδομένων από ιστοσελίδα με κριτικές ξενοδοχείων. Χρησιμοποιήσαμε επίσης για τις δοκιμές μας το πλέον standard σύνολο δεδομένων για την ανάλυση άποψης, το σύνολο κριτικών ταινιών των Pang/Lee [PL04]. Τέλος, για την εφαρμογή του μοντέλου Hidden Markov, χρειάστηκε να σημειώσουμε χειρονακτικά συγκεκριμένες λέξεις και φράσεις και να τις κατατάξουμε σε θεματικές κατηγορίες. Αυτό έγινε σε μέρος των κειμένων που συγκεντρώσαμε από την ιστοσελίδα κριτικών ξενοδοχείων.

4.1.1 Θετικά-αρνητικά στοιχεία ξενοδοχείων

Επειδή η ακριβής μορφή του συνόλου δεδομένων επηρεάζει ποιοτικά τα πειράματα που γίνονται πάνω σε αυτό, εδώ θα αναφέρουμε λίγα στοιχεία για το dataset αυτό. Στην ιστοσελίδα οι χρήστες, αφού έχουν επισκεφθεί και διαμείνει σε ένα ξενοδοχείο, καλούνται να αναφέρουν τα θετικά και τα αρνητικά στοιχεία από τη διαμονή τους, σε χωριστά κείμενα.

Το σύνολο δεδομένων που συλλέξαμε περιείχε 5805 κείμενα που αναφέρονταν στα θετικά στοιχεία και 3645 κείμενα που αναφέρονταν στα αρνητικά. Ο λόγος που το πλήθος των κειμένων είναι διαφορετικό στις δύο κατηγορίες είναι ότι αρκετοί χρήστες επέλεγαν να συμπληρώσουν μόνο το κείμενο της μιας κατηγορίας (και όταν συνέβαινε αυτό, συνήθως συμπλήρωναν μόνο τα θετικά). Όλα τα κείμενα περιείχαν κριτική στην ελληνική γλώσσα, αλλά ενδεχομένως και αγγλικές λέξεις. Τα ξενοδοχεία τα οποία αφορούσαν οι κριτικές ήταν όλα ξενοδοχεία της περιοχής της Αττικής.

Παράδειγμα αναφοράς των θετικών στοιχείων από χρήστη φαίνεται στο σχήμα 4.1 ενώ παράδειγμα αναφοράς αρνητικών στοιχείων στο σχήμα 4.2. Επίσης, παραδείγματα του πως φαίνονταν τα σχόλια στην ίδια την ιστοσελίδα φαίνονται στα σχήματα 4.3 και 4.4.

Χαρακτηριστικό των κειμένων που συγκεντρώθηκαν είναι η σαφήνεια και η περιεκτικότητα. Γενικά όταν οι χρήστες γράφουν κριτικές προϊόντων, όπου ο σκοπός της κριτικής είναι να βοηθήσουν μελλοντικούς χρήστες να αποφασίσουν εάν αξίζει να αγοράσουν αυτό το προϊόν, αναφέρονται με απλή γλώσσα και αμεσότητα σε εκφάνσεις του προϊόντος, εάν αυτές ήταν ικανοποιητικές ή όχι, εάν υπήρχαν ελλείψεις ή εάν παρουσιάστηκαν προβλήματα. Εάν είναι ικανοποιημένοι, το δηλώνουν συχνά ξεκάθαρα. Ο θεματικός τομέας των ξενοδοχείων παρουσιάζει αυτά τα χαρακτηριστικά. Θα μπορούσαμε να πούμε ότι οι κριτικές των ξενοδοχείων είναι από αυτή την άποψη παρόμοιες με κριτικές π.χ. για ηλεκτρικές ή ηλεκτρονικές συσκευές.

Λόγω αυτών των χαρακτηριστικών, στο θεματικό τομέα των ξενοδοχείων αυτόματες μέθοδοι ανάλυσης άποψης μπορούν να δουλέψουν με μεγάλη αποτελεσματικότητα. Αντίθετα, σε άλλες κατηγορίες κειμένων όπου τίθεται θέμα θετικής/αρνητικής άποψης, όπως κριτικές ταινιών, βιβλίων ή κείμενα με θέμα την πολιτική, οι χρήστες εκφράζονται με περισσότερα λόγια και με πιο πολύπλοκο τρόπο, αφού το ζητούμενο αυτών των κειμένων είναι βαθύτερο από το αν ένα προϊόν είναι καλό ή κακό, με συνέπεια να είναι πιο δύσκολο οι αυτόματες μέθοδοι να πετύχουν το σωστό αποτέλεσμα. Το επόμενο σύνολο δεδομένων που χρησιμοποιήσαμε ανήκει σ᾽ αυτή την κατηγορία, αφού αφορά στις κριτικές ταινιών.

4.1.2 Κριτικές ταινιών

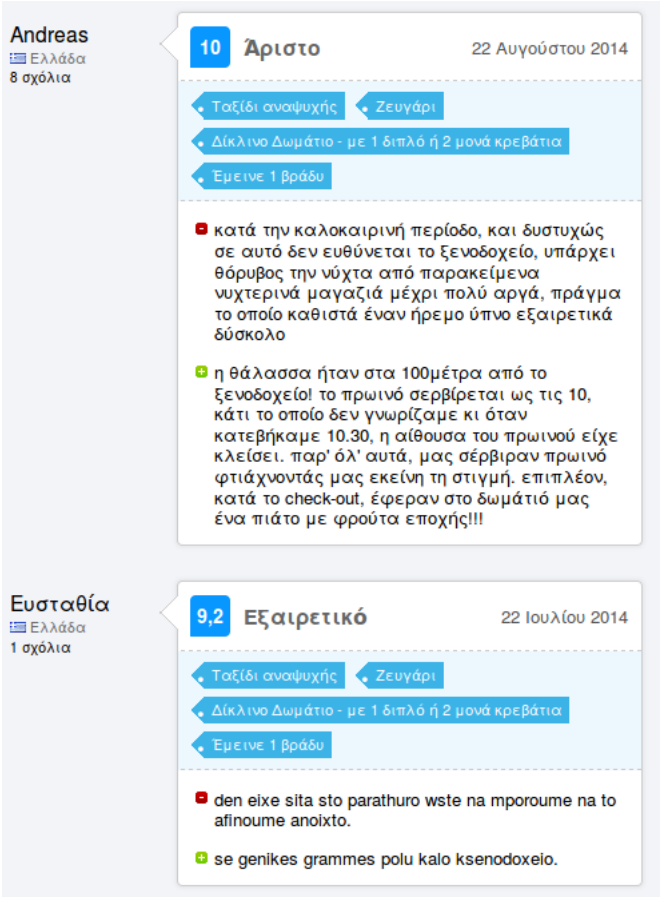
Το σύνολο κριτικών ταινιών των Bo Pang και Lillian Lee χρησιμοποιείται σε πολλές δημοσιεύσεις που αναπτύσσουν τεχνικές μηχανικής μάθησης για

“ Το ξενοδοχείο δεν είναι πλήρως ανακαινισμένο, χωρίς όμως να υπάρχει κάποιο πρόβλημα. Το μπάνιο είναι σαν καινούριο. Πάρα πολύ καλη εξυπηρετηση και πάρα πολύ ευγενικό προσωπικό. Πολύ καλό πρωϊνό. Θα επιθυμούσα λίγο καλυτερη τιμή, αλλά σε γενικές γραμμές, θα το επισκεπτόμουν πάλι ευχαρίστως. ”

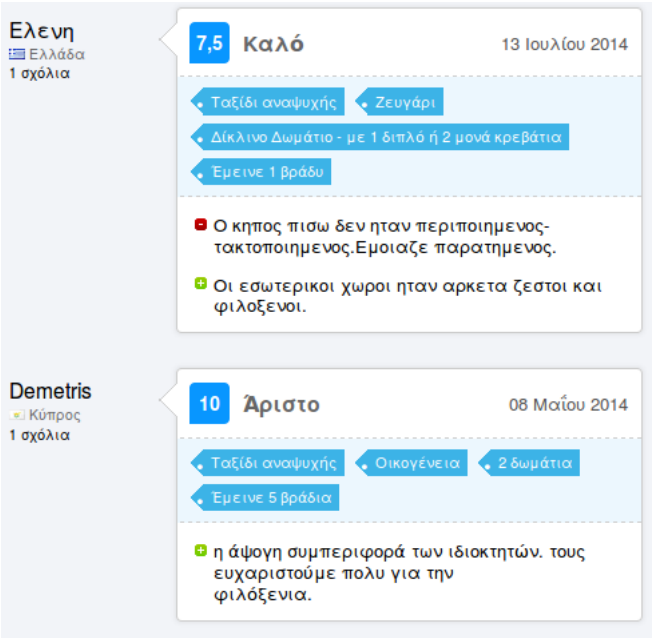
Σχήμα 4.1: Παράδειγμα αναφοράς θετικών στοιχείων

“ Πολύ μικρό ασανσερ. Το μπαλκόνι του έβλεπε σε έναν απελπιστικά ασφυκτικό, κλειστό ακάλυπτο. Για την ποιότητα του δωματίου ακόμα κι η τιμή της προσφοράς θεωρώ ότι ήταν μεγάλη. ”

Σχήμα 4.2: Παράδειγμα αναφοράς αρνητικών στοιχείων



Σχήμα 4.3: Screenshot ιστοσελίδας 1



Σχήμα 4.4: Screenshot ιστοσελίδας 2

ανάλυση άποψης ως ένα κοινό σημείο αναφοράς ώστε να συγκρίνεται η επίδοση των διάφορων συστημάτων. Από τους δύο συγγραφείς διατίθενται μια σειρά από σύνολα δεδομένων. Τα δύο πιο ενδιαφέροντα για τη δικιά μας χρήση ήταν τα παρακάτω:

- Polarity dataset v2.0: Περιέχει 1000 θετικές και 1000 αρνητικές κριτικές ταινιών. Πρωτοδημοσιεύθηκε το 2004. Οι κριτικές είναι σχετικά μακροσκελείς, συνήθως μήκους μερικών μεγάλων παραγράφων, δηλαδή το μέγεθός τους ανέρχεται στις εκατοντάδες λέξεις.

- Sentence polarity dataset v1.0: Περιέχει 5331 θετικές και 5331 αρνητικές προτάσεις. Αυτές οι προτάσεις περιέχονταν στις κριτικές ταινιών. Είναι ουσιαστικά περίοδοι (από τελεία σε τελεία), και το μήκος τους κυμαίνεται από λιγότερες των 10 λέξεων μέχρι μερικές δεκάδες λέξεις. Το dataset πρωτοδημοσιεύτηκε το 2005.

Η χρήση αυτού του συνόλου δεδομένων έγινε γιατί όπως είπαμε πάρα πολλοί έχουν ήδη χρησιμοποιήσει αυτό το σύνολο οπότε οι ερευνητές μπορούν να συγκρίνουν τα νούμερα απόδοσής τους με αυτά άλλων. Σημειώνεται ότι μεταξύ των ερευνητών που χρησιμοποιούν αυτό το σύνολο δεδομένων περιλαμβάνονται και φοιτητές, τόσο της Σχολής ΗΜΜΥ όσο και άλλων ελληνικών πανεπιστημίων, σε διπλωματικές εργασίες που αναφέρονται στον τομέα της ανάλυσης άποψης.

Ως προς τα χαρακτηριστικά των κειμένων αυτών, πρόκειται για κείμενα που ανήκουν στην κατηγορία των κριτικών έργων τέχνης. Όπως αναφέραμε και στην προηγούμενη ενότητα, τέτοιες κριτικές δεν περιορίζονται στην απλή αποτίμηση ενός προϊόντος, αλλά συχνά οι χρήστες αναπτύσσουν πιο σύνθετες και μακροσκελείς σκέψεις και δεν εκφράζουν την κρίση τους απλά και άμεσα.

4.2 Εκτίμηση της αποτελεσματικότητας

Για το σκοπό της σύγκρισης της αποτελεσματικότητας διαφορετικών μοντέλων και των παραμέτρων τους, χρειάζεται να έχουμε κάποια αριθμητικά μέτρα της απόδοσης τους, εννοώντας τη δυνατότητα να επιτελούν το σκοπό τους αποτελεσματικά. Για να θεωρηθεί ότι ένα σύστημα δουλεύει καλά, πρέπει να ταξινομεί σωστά παραδείγματα τα οποία δεν έχει δει στο παρελθόν [RN].

4.2.1 Διαχωρισμός συνόλων εκπαίδευσης–ελέγχου

Ο πιο απλός τρόπος να ελεγχθεί η απόδοση είναι τα ταξινομημένα δεδομένα που διαθέτουμε να μοιραστούν σε δύο χωριστά σύνολα: το σύνολο εκπαίδευσης (training set) και το σύνολο ελέγχου (test set). Το μοντέλο τότε εκπαιδεύεται με βάση τα δεδομένα του συνόλου εκπαίδευσης, και λαμβάνουμε τις εκτιμήσεις του για τα παραδείγματα του συνόλου ελέγχου. Συγκρίνοντας τις εκτιμήσεις του μοντέλου με τις σωστές απαντήσεις, λαμβάνουμε το μέτρο της απόδοσης του μοντέλου. Σ' αυτή την περίπτωση το σύνολο ελέγχου καλείται και αποκλειστικό σύνολο ελέγχου (dedicated test set).

4.2.2 Η μέθοδος k-fold cross-validation

Η προηγούμενη μέθοδος έχει το ελάττωμα πως η μετρούμενη απόδοση μπορεί να μεταβάλλεται με βάση το ποια στοιχεία επιλέχθηκαν για να ανήκουν στο test set. Μια εναλλακτική μέθοδος πετυχαίνει να χρησιμοποιούνται όλα τα παραδείγματα του διαθέσιμου συνόλου δεδομένων ως δεδομένα εκπαίδευσης και δεδομένα ελέγχου, σε διαφορετικές φάσεις, εξασφαλίζοντας όμως ότι στο ίδιο πείραμα δεν θα χρησιμοποιηθούν γνωστά στο μοντέλο δεδομένα ως δεδομένα ελέγχου. Αυτή η μέθοδος ονομάζεται κ-μερής σταυρωτός έλεγχος (k-fold cross-validation) και λειτουργεί ως εξής:

- Επιλέγεται θετικός ακέραιος K, και το σύνολο δεδομένων χωρίζεται σε K κομμάτια ίσου μεγέθους.
- Γίνονται K ξεχωριστά πειράματα. Σε κάθε ένα από αυτά, ένα από τα K κομμάτια του συνόλου δεδομένων παίζει το ρόλο του συνόλου ελέγχου,

ενώ τα υπόλοιπα συνιστούν το σύνολο εκπαίδευσης. Σε κάθε πείραμα μετριέται το μέτρο της απόδοσης.

- Αφού γίνουν οι K δοκιμές, το συνολικό μέτρο απόδοσης είναι ο μέσος όρος των μέτρων που προέκυψαν.

Έτσι η βασική αρχή του διαχωρισμού δεδομένων εκπαίδευσης και ελέγχου τηρείται, όμως όλα τα διαθέσιμα παραδείγματα χρησιμοποιούνται και στους δύο ρόλους (και συγκεκριμένα, κάθε παράδειγμα συμμετέχει $K-1$ φορές στο σύνολο εκπαίδευσης και μια φορά στο σύνολο ελέγχου). Έτσι, τυχαίες επιδράσεις στην απόδοση λόγω ενός συγκεκριμένου διαχωρισμού αποφεύγονται και η μεταβλητότητα στο αποτέλεσμα εξομαλύνεται.

Για την επιλογή της τιμής του K, πρέπει να προσεχθεί αυτή να μην είναι τόσο μεγάλη ώστε το μέγεθος του συνόλου ελέγχου να είναι πολύ μικρό (αυτό έχει μέγεθος το $1/K$ των συνολικών δεδομένων). Επίσης, όσο μεγαλύτερη είναι η τιμή του K τόσο περισσότερο χρόνο θα διαρκεί η εκτέλεση των δοκιμών, αφού γίνονται K δοκιμές. Συνήθως χρησιμοποιούνται τιμές στο εύρος 3-10, με το $K = 10$ να προτιμάται συχνά. Μάλιστα έχει παρατηρηθεί ότι δεν αξίζει το K να είναι μεγαλύτερο από 10 ακόμα κι αν αυτό είναι υπολογιστικά εφικτό [Koh+95].

4.2.3 Οι μετρικές της αποτελεσματικότητας

Precision και Recall

Δύο πολύ βασικές μετρικές απόδοσης σε προβλήματα κατηγοριοποίησης είναι η ακρίβεια (precision) και η ανάκληση (recall). Αυτές οι δύο μετρικές δεν λαμβάνουν μια μοναδική τιμή για ένα πείραμα, αλλά λαμβάνουν μια τιμή για κάθε κατηγορία. Για παράδειγμα, έστω ένα πρόβλημα κατηγοριοποίησης κειμένων όπου υπάρχει μια κατηγορία c. Μετά από χρήση του μοντέλου σε ένα σύνολο ελέγχου, η ακρίβεια και η ανάκληση για την κατηγορία c ορίζονται ως εξής:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

όπου

- TP: True Positives: Ο αριθμός των κειμένων που ανήκαν στην κατηγορία c, και τοποθετήθηκαν σωστά στην κατηγορία c από το μοντέλο.

- FP: False Positives: Ο αριθμός των κειμένων που δεν ανήκαν στην κατηγορία c, και τοποθετήθηκαν λανθασμένα στην κατηγορία c από το μοντέλο.

- FN: False Negatives: Ο αριθμός των κειμένων που ανήκαν στην κατηγορία c, και τοποθετήθηκαν λανθασμένα σε άλλη κατηγορία από το μοντέλο.
- TN: True Negatives: Ο αριθμός των κειμένων που δεν ανήκαν στην κατηγορία c, και τοποθετήθηκαν σε κάποια κατηγορία διαφορετική της c από το μοντέλο. Αυτή η ποσότητα δεν αναφέρεται στους δύο παραπάνω τύπους αλλά την αναφέρουμε για λόγους πληρότητας. Παρατηρήστε ότι στις περιπτώσεις αυτές δε σημαίνει ότι το μοντέλο έκανε απαραίτητα τη σωστή επιλογή, αλλά αυτό δε μας αφορά όταν ορίζουμε το Precision και το Recall για μια συγκεκριμένη κατηγορία.

Ο παραπάνω ορισμός της ακρίβειας και της ανάκλησης δουλεύει και όταν οι κατηγορίες είναι περισσότερες από δύο. Η ακρίβεια δίνει το μέτρο του κατά πόσο τα κείμενα που κατηγοριοποιούνται σε μια κατηγορία είναι όντως μέλη αυτής της κατηγορίας, αγνοώντας το πόσα πραγματικά κείμενα της κατηγορίας δεν εντοπίστηκαν. Η ανάκληση δίνει το μέτρο του πόσα από όλα τα κείμενα της εν λόγω κατηγορίας εντοπίστηκαν σωστά (ανακλήθηκαν) από το μοντέλο, αγνοώντας όμως τυχόν κείμενα που λανθασμένα ταξινομήθηκαν στην ίδια αυτή κατηγορία.

Το πόσο μας ενδιαφέρει η μία ή η άλλη από αυτές τις δύο μετρικές εξαρτάται από τις ανάγκες της εφαρμογής μας. Για παράδειγμα, μπορεί να είναι επιθυμητό να εμφανίζονται κείμενα με θετική γνώμη αυτόματα σε μια ιστοσελίδα, και να ενδιαφέρει τα όσα κείμενα επιλέγονται να είναι όντως θετικά. Εκεί ενδιαφέρει η ακρίβεια και όχι η ανάκληση. Σε άλλο παράδειγμα, μπορεί να θέλουμε να δούμε όλα τα αρνητικά σχόλια για ένα προϊόν, ενδεχομένως για να ανταποκριθούμε σε αυτά. Αν ενδιαφέρει να μη μείνει κάποιο αρνητικό σχόλιο αναπάντητο, τότε θέλουμε υψηλή ανάκληση για τα αρνητικά σχόλια και δε μας αφορά τόσο η ακρίβεια σε αυτά, π.χ. αν δε μας πειράζει τόσο το να προσπεράσουμε κάποια θετικά σχόλια λαθεμένα ταξινομημένα ως αρνητικά.

Είναι εύκολο να συμπεράνει κανείς ότι υπάρχει μια αντίστροφη σχέση μεταξύ precision και recall. Ένα μοντέλο που κάνει συντηρητικές επιλογές ως προς μια κατηγορία μπορεί να πετύχει μεγάλο precision και μικρό recall, ή το αντίστροφο εάν το μοντέλο είναι προκατειλημμένο (biased) προς μια κατηγορία, και αποδίδοντάς την πολύ συχνά, έχει μικρό precision αλλά μεγαλύτερο recall σε αυτή.

F-Score

Οι δύο αυτές μετρικές μπορούν να συνδυαστούν στο F-Measure (ή F-Score), που είναι ο αρμονικός μέσος των δύο, ως εξής:

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Η γενικότερη μορφή του παραπάνω συνδυαστικού μέτρου είναι το F_β Measure, που ορίζεται ως εξής:

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

Για $\beta = 1$ λαμβάνουμε το απλό F-Measure, που για το λόγο αυτό λέγεται και F_1 score. Αν επιλέξουμε $\beta > 1$, η ανάκληση έχει μεγαλύτερο βάρος από την ακρίβεια, ενώ το αντίθετο συμβαίνει για $\beta > 1$. Έτσι, τα μέτρα F_2 και $F_{0.5}$ χρησιμοποιούνται αν θέλουμε να δοθεί περισσότερη έμφαση στο recall ή στο precision αντίστοιχα.

Παρότι το F-Score είναι βολικό γιατί η αποτελεσματικότητα για κάθε κατηγορία αποδίδεται σε ένα μόνο νούμερο, πρέπει να ελέγχουμε την ακρίβεια και την ανάκληση που το μοντέλο μας αποδίδει, για να είμαστε σίγουροι ότι δεν παρουσιάζει ανεπιθύμητη συμπεριφορά. Συχνά κατά την ανάπτυξη ενός μοντέλου μπορεί από λάθος το μοντέλο να επιδεικνύει προκατάληψη προς μια κατηγορία, με μεγάλο recall και μικρό precision, ή το αντίθετο, εάν μια κατηγορία επιλέγεται σπάνια από το μοντέλο. Αυτά τα στοιχεία φανερώνονται μόνο από τον έλεγχο των τιμών precision και recall για κάθε κατηγορία.

Accuracy

Έχοντας προειδοποιήσει για τη σημασία του να ελέγχονται η ανάκληση και η ακρίβεια για καθεμιά από τις κατηγορίες, η πιο δημοφιλής μετρική είναι γενικά η ορθότητα (accuracy). Όταν έχουμε έναν αριθμό από άγνωστα κείμενα και το μοντέλο μας κατατάξει το καθένα από αυτά σε μια κατηγορία, η ορθότητα είναι απλά ο λόγος των σωστών επιλογών του μοντέλου προς το συνολικό πλήθος των κειμένων. Το ποσοστό της ορθότητας είναι ένα νούμερο που χαρακτηρίζει ένα πείραμα συνολικά. Στην περίπτωση της δυαδικής ταξινόμησης (μόνο δύο κατηγορίες), για να θεωρείται ένα μοντέλο ικανοποιητικό, απαραίτητη προϋπόθεση είναι η ορθότητα του να ξεπερνά αρκετά το 50%. Αυτό γιατί ορθότητα κοντά στο 50% μπορεί να επιτευχθεί είτε από ένα πολύ προκατειλημμένο μοντέλο, που αποδίδει σχεδόν συνεχώς την ίδια κατηγορία (οπότε είναι σωστό περίπου τις μισές φορές), είτε από ένα μοντέλο που αποδίδει κατηγορία σχεδόν τυχαία.

4.3 Naive Bayes

4.3.1 Δοκιμή σε κριτικές ξενοδοχείων

Το μοντέλο Naive Bayes δοκιμάστηκε πρώτα στο σύνολο δεδομένων των κριτικών ξενοδοχείων. Εκτελέστηκαν δοκιμές χρησιμοποιώντας τη μέθοδο 10-way cross-validation. Για το χωρισμό του κειμένου σε λέξεις, αρχικά έγινε διάσπαση όπου υπήρχαν κενά ή περίεργοι χαρακτήρες (που δεν ανήκαν στο σύνολο των ελληνικών και αγγλικών αλφαβήτων, αλλά ούτε ήταν ψηφία ή σημεία στίξεως) οι οποίοι και αγνοούνταν. Στις λέξεις που απέμεναν, εφαρμοζόταν

μια δεύτερη διαδικασία χωρισμού εάν αυτές περιείχαν κάποιο σημείο στίξης. Αυτό διαχωριζόταν από τη λέξη και διατηρούταν ως μια λέξη από μόνο του. Τέλος, οι λεκτικές μονάδες κανονικοποιούνταν ως προς τα κεφαλαία και τους τόνους.

Στη συνέχεια, εφαρμόστηκαν κάποιες παραλλαγές στην περαιτέρω διαμόρφωση των χαρακτηριστικών που τελικά θα χρησιμοποιούσε το μοντέλο Naive Bayes. Αυτές ήταν το αν θα χρησιμοποιούνται μόνο μεμονωμένες λέξεις ως χαρακτηριστικά, η και bigrams (ζεύγη διαδοχικών λέξεων) ως χαρακτηριστικά. Επίσης, το αν σπάνιες λέξεις (που εμφανίζονταν μόνο μια φορά στο σύνολο εκπαίδευσης) θα αγνοούνταν. Τέλος, δοκιμάστηκε οι λέξεις, τόσο σαν μεμονωμένες όσο και σαν μέρος bigram να αντικαθίστανται από το stem τους, για ευρύτερη κανονικοποίηση.

Στον πίνακα 4.1 παρουσιάζονται τα ποσοστά ακρίβειας και ανάκλησης ανά κατηγορία (θετική, αρνητική γνώμη) και το ποσοστό ορθότητας συνολικά, για τις διαφορετικές παραλλαγές του μοντέλου Naive Bayes, ως προς την επιλογή των χαρακτηριστικών με τα οποία αναπαρίστονταν τα κείμενα στο μοντέλο.

Χαρακτηριστικά	Θετικά σχόλια		Αρνητικά σχόλια		Συνολικά
	Precision	Recall	Precision	Recall	Accuracy
λέξεις	95.1%	95.3%	92.6%	92.2%	94.1%
λέξεις συχνότητα >= 2	94.9%	95.3%	92.5%	91.9%	94.0%
λέξεις (stems)	94.9%	95.2%	92.3%	91.9%	93.9%
λέξεις + bigrams	94.9%	95.9%	93.3%	91.9%	94.3%
λέξεις + bigrams συχνότητα >= 2	94.7%	96.0%	93.5%	91.4%	94.2%
λέξεις + bigrams (stems)	95.3%	95.8%	93.2%	92.5%	**94.5%**

Πίνακας 4.1: Αποτελέσματα παραλλαγών Naive Bayes—Σύνολο κριτικών ξενοδοχείων

Το πιο πετυχημένο είδος χαρακτηριστικών είναι η χρήση του stemming, μαζί με το συνδυασμό λέξεων και bigrams, με ποσοστό ορθότητας 94.5%.

Παρατηρούμε ιδιαίτερα υψηλά ποσοστά ορθότητας σε επίπεδα περίπου 94% για όλες τις παραλλαγές του μοντέλου, γεγονός που βασικά δηλώνει ότι το dataset μας είναι "εύκολο". Αυτό προκύπτει λογικά αν σκεφτούμε τον τρόπο με τον οποίο δημιουργήθηκε το dataset. Συχνά όταν δημιουργούνται σύνολα δεδομένων με κριτικές, αυτές οι κριτικές περιέχουν και τα αρνητικά και τα θετικά στοιχεία του εν λόγω αντικειμένου στο ίδιο κείμενο. Αν οι κριτικές είναι να μοιραστούν σε δύο μόνο κατηγορίες, θετικές και αρνητικές, αυτό γίνεται συχνά βάσει του score που συνοδεύει την κριτική, και επιλέγεται ένα κατώφλι, πάνω από το οποίο η κριτική θεωρείται θετική, και κάτω από αυτό

αρνητική. Έτσι, σε κριτικές που κατηγοριοποιούνται ως θετικές περιέχονται και αρνητικές παρατηρήσεις, αλλά με το θετικό στοιχείο τελικά να υπερισχύει. Αντίστοιχα και για τις αρνητικές κριτικές. Και έτσι το μοντέλο μπορεί να αποτύχει να "αντιληφθεί" ποια άποψη υπερισχύει στο κείμενο.

Το δικό μας σύνολο δεδομένων όμως συλλέχτηκε από ιστοσελίδα όπου οι χρήστες καλούνταν να αναφέρουν τα θετικά και τα αρνητικά τους σχόλια σε δύο χωριστά κείμενα. Έτσι, τα κείμενά μας είναι πιο "καθαρά" αρνητικά ή θετικά από αυτά στα περισσότερα σύνολα δεδομένων. Αυτό το στοιχείο αφενός βοηθάει το μοντέλο κατά την εκπαίδευση να μάθει καλύτερα τι είναι θετικό και τι αρνητικό σχόλιο. Αφετέρου, και κυριότερα, κατά τη διαδικασία ελέγχου σε άγνωστα κείμενα, δεν ταξινομεί άγνωστες κριτικές που περιέχουν ενδεχομένως μείγμα θετικών και αρνητικών παρατηρήσεων, αλλά ταξινομεί κείμενα που είτε αναφέρονται μόνο στα θετικά του ξενοδοχείου, είτε μόνο στα αρνητικά, διεργασία που είναι πιο εύκολη.

Βέβαια, όπως προαναφέραμε και στην υποενότητα 4.1.1, ούτως ή άλλως όταν οι χρήστες γράφουν κριτικές για ξενοδοχεία είναι άμεσοι και περιεκτικοί στο λόγο τους, οπότε αυτό βοηθά να υπάρχει μεγάλο Accuracy.

Ένας ακόμα λόγος για το μεγάλο ποσοστό ορθότητας είναι το ικανοποιητικό μέγεθος του συνόλου δεδομένων, ειδικά σε σχέση με το ότι στις κριτικές ξενοδοχείων αρκετές παρατηρήσεις αναφέρονται ξανά και ξανά (π.χ. καθαριότητα), λόγω του ότι η διαμονή σε ξενοδοχείο παρουσιάζει μικρότερη μεταβλητότητα στο εύρος των εμπειριών, σε σχέση π.χ. με μια ταινία.

Μια ακόμα παρατήρηση που μπορεί να γίνει είναι ότι το μοντέλο υπερτερεί λίγο στα θετικά σχόλια σε σχέση με τα αρνητικά, με την ακρίβεια στα θετικά να είναι περίπου 1-2 ποσοστιαίες μονάδες μεγαλύτερη και την ανάκληση στα θετικά να είναι επίσης περίπου 3-4 ποσοστιαίες μονάδες μεγαλύτερη, σε όλες τις παραλλαγές του μοντέλου. Η απόκλιση αυτή όμως δεν είναι πολύ μεγάλη ώστε να μιλάμε για προβληματικά προκατειλημμένο μοντέλο, αφού τα ποσοστά είναι υψηλά και στις δύο κατηγορίες.

Επισκόπηση παραμέτρων που μαθεύτηκαν

Το μοντέλο Naive Bayes δίνει τη δυνατότητα επισκόπησης των παραμέτρων που έχει μάθει. Στον πίνακα 4.2 φαίνονται οι 30 λέξεις πιο ενδεικτικές για αρνητικό και θετικό σχόλιο:

Στις λέξεις του μοντέλου εφαρμόστηκε κανονικοποίηση ως προς τα κεφαλαία και τους τόνους, γι αυτό και οι λέξεις είναι με μικρά και άτονες. Παρατηρώντας τις λέξεις μπορούμε να σταθούμε σε μερικά σημεία:

- Οι περισσότερες λέξεις είναι άμεσα ενδεικτικές αρνητικής ή θετικής άποψης. Κυρίως επίθετα όπως π.χ. "απαράδεκτο" και "κακή" για τα αρνητικά ή "καταπληκτική" και "εξαιρετικό" για τα θετικά, αλλά και άλλα μέρη του λόγου με ξεκάθαρη πολικότητα όπως "μπράβο".

Οι πιο αρνητικές λέξεις	
Λέξη	Πιθανότητα θετικού σχολίου
φτωχο	0.0116
απαραδεκτο	0.0140
κακο	0.0149
βρωμικο	0.0164
ελλειψη	0.0168
ανυπαρκτη	0.0181
κακη	0.0217
περισσοτερη	0.0236
υγρασια	0.0253
κουρτινα	0.0256
αβολο	0.0296
κουρτινες	0.0296
αποτελεσμα	0.0300
τουαλετας	0.0324
χρηζει	0.0340
ανυπαρκτο	0.0358
ασχημη	0.0358
μοκετα	0.0374
αδιαφορο	0.0377
εντονη	0.0377
καζανακι	0.0377
κακης	0.0377
σκληρα	0.0377
ακουγοταν	0.0399
βρωμικα	0.0399
βρωμικη	0.0399
διαδρομους	0.0399
πετσετα	0.0399
σαπουνι	0.0399
παραθυρα	0.0410

Οι πιο θετικές λέξεις	
Λέξη	Πιθανότητα θετικού σχολίου
ευγενεστατο	0.9910
ησυχη	0.9899
ησυχο	0.9895
ευγενεια	0.9886
εξαιρετικο	0.9872
αψογο	0.9867
αριστη	0.9845
ευγενικο	0.9838
φιλοξενοι	0.9833
φιλικο	0.9813
ωραια	0.9804
φοβερη	0.9800
φιλικη	0.9787
περασαμε	0.9778
καταπληκτικη	0.9757
φιλοξενια	0.9744
εξυπηρετικοι	0.9744
ευγενεστατοι	0.9744
εξυπηρετικο	0.9740
ανεπιφυλακτα	0.9739
εξαιρετικη	0.9736
προτεινω	0.9722
φανταστικη	0.9722
φιλικοτητα	0.9722
φιλοξενο	0.9721
διακοσμημενο	0.9706
μπραβο	0.9706
κηπος	0.9677
προθυμια	0.9677
ιδιοκτητες	0.9676

Πίνακας 4.2: Πιο ενδεικτικές λέξεις—κριτικές ξενοδοχείων

- Άλλες λέξεις είναι επίσης ξεκάθαρης πολικότητας, όμως σχετίζονται με τη θεματική κατηγορία των ξενοδοχείων, όπως "βρώμικο" στα αρνητικά και "ευγενέστατο", "φιλόξενοι" στα θετικά.

- Όμως άλλες λέξεις δεν έχουν προφανή πολικότητα αν ειδωθούν μεμονωμένα. Αυτές είναι για παράδειγμα οι θετικές λέξεις "περάσαμε", "κήπος" και "ιδιοκτήτες", και οι αρνητικές "κουρτίνα", "μοκέτα", "διαδρόμους", "ακουγόταν", "αποτέλεσμα" και "παράθυρα". Αυτές οι λέξεις ονομάζουν στοιχεία που εμφανίζονται στα σχόλια των χρηστών δυσανάλογα όταν μιλούν για τα θετικά ή τα αρνητικά, δηλαδή π.χ. οι επισκέπτες αναφέρονται στη μοκέτα του δωματίου μόνο όταν κάτι δεν πάει καλά με αυτή, ή αναφέρονται στον κήπο συνήθως σαν κάτι θετικό. Επίσης, είναι μέρος εκφράσεων που πιθανώς χρησιμοποιούνται συχνότερα στην περίπτωση του αρνητικού ("με αποτέλεσμα") ή του θετικού σχολίου ("περάσαμε καλά").

Το νόημα του σχολιασμού μας δεν είναι να εξηγήσουμε γιατί κάθε λέξη μπορεί να είναι ενδεικτική θετικής/αρνητικής άποψης, αλλά να παρατηρήσουμε ότι το μοντέλο Naive Bayes ανακαλύπτει ποικίλες προφανείς ή μη συσχετίσεις και τις χρησιμοποιεί για να κατατάξει ένα άγνωστο κείμενο. Επίσης, επιβεβαιώνεται ένα κεντρικό θέμα της μηχανικής μάθησης, ότι το τι μαθαίνει το μοντέλο εξαρτάται πάρα πολύ από τα δεδομένα εκπαίδευσης, και πρέπει να χρησιμοποιείται στη συνέχεια σε δεδομένα από τον ίδιο θεματικό τομέα. Έτσι, ενώ οι παραπάνω παράμετροι που έμαθε το μοντέλο μπορεί να έχουν νόημα για κριτικές ξενοδοχείων, δε γενικεύονται το ίδιο καλά όλες. Η λέξη "περάσαμε" δεν είναι γενικά θετική, ούτε η λέξη "παράθυρα" αρνητική.

Μια ακόμα παρατήρηση που μπορεί να γίνει είναι ότι ένα μοντέλο Naive Bayes μπορεί να χρησιμεύσει ως σημαντικό βοήθημα για την κατάρτιση μιας πιο γενικής λίστας αρνητικών/θετικών λέξεων, με την ακόλουθη διαδικασία: Μετά την εκπαίδευση του μοντέλου, επιλέγονται χειρονακτικά οι μη γενικές θετικές ή αρνητικές λέξεις και απομακρύνονται από τη λίστα. Αυτές που μένουν είναι μια καλή λίστα για γενικότερη χρήση, όχι μόνο για το θεματικό τομέα των κειμένων εκπαίδευσης, που συγκεντρώθηκε εν μέρει αυτοματοποιημένα.

Στον πίνακα 4.3 φαίνονται παρομοίως τα 30 πιο ενδεικτικά bigrams (ζεύγη διαδοχικών λέξεων) που έμαθε το μοντέλο. Παρατηρούμε ότι αυτά είναι πολύ περισσότερο εξαρτώμενα από τη θεματική κατηγορία των κειμένων εκπαίδευσης.

4.3.2 Δοκιμή σε κριτικές ταινιών

Στη συνέχεια το μοντέλο Naive Bayes δοκιμάστηκε στο κλασικό σύνολο δεδομένων του πεδίου της ανάλυσης άποψης, το σύνολο κριτικών ταινιών των Pang και Lee. Η δοκιμή αυτή μας επιτρέπει να επιβεβαιώσουμε ότι το μοντέλο μας λειτουργεί και σε ένα πιο απαιτητικό σύνολο δεδομένων.

Τα πιο αρνητικά bigrams	
Bigram	Πιθανότητα θετικού σχολίου
φτωχο πρωινο	0.0072
μπανιο δεν	0.0179
μικρο δωματιο	0.0188
κακη ηχομονωση	0.0194
του μπανιου	0.0243
πολυ μικρο	0.0256
μικρα δωματια	0.0258
δεν δουλευε	0.0278
απο δρομο	0.0289
πορτα του	0.0289
πολυ φτωχο	0.0293
πολυ θορυβο	0.0302
μικρο μπανιο	0.0317
τηλεοραση δεν	0.0346
της τουαλετας	0.0346
ηταν πιο	0.0364
ξενοδοχειο δεν	0.0370
δεν ειχε	0.0371
δεν υπηρχε	0.0376
απαραδεκτο για	0.0383
αρκετη φασαρια	0.0406
δωματιο μυριζε	0.0406
ουτε καν	0.0406
πολυ παλιο	0.0406
δεν αξιζει	0.0430
κακης ποιοτητας	0.0430
καλο ηταν	0.0430
περιοχη δεν	0.0430
στο πατωμα	0.0430
θορυβος απο	0.0443

Τα πιο θετικά bigrams	
Bigram	Πιθανότητα θετικού σχολίου
ευγενικο προσωπικο	0.9952
πολυ ωραια	0.9935
και εξυπηρετικο	0.9933
φιλικο προσωπικο	0.9929
πολυ ευγενικο	0.9925
ευγενικο και	0.9912
πολυ φιλικο	0.9907
καλη τοποθεσια	0.9900
υπεροχη θεα	0.9878
καλη εξυπηρετηση	0.9872
ευγενεια του	0.9861
καθαρα δωματια	0.9855
ιδανικο για	0.9851
εξαιρετικη τοποθεσια	0.9846
πολυ ομορφο	0.9844
ανετα δωματια	0.9839
αψογη εξυπηρετηση	0.9839
τοποθεσια πολυ	0.9836
αριστη σχεση	0.9831
συνιστω ανεπιφυλακτα	0.9828
εξυπηρετικο προσωπικο	0.9823
προσωπικο πολυ	0.9820
ωραιο πρωινο	0.9818
πολυ καθαρα	0.9811
ωραια τοποθεσια	0.9811
πολυ εξυπηρετικο	0.9810
αριστη εξυπηρετηση	0.9796
ευγενεια και	0.9796
κεντρικο σημειο	0.9792
και φιλικο	0.9783

Πίνακας 4.3: Πιο ενδεικτικά bigrams—κριτικές ξενοδοχείων

Τα αποτελέσματα των δοκιμών, πάλι 10-fold cross-validation, για μερικές παραλλαγές του μοντέλου φαίνονται στον πίνακα 4.4.

Χαρακτηριστικά	Θετικές κριτικές		Αρνητικές κριτικές		Συνολικά
	Precision	Recall	Precision	Recall	Accuracy
λέξεις	91.6%	59.0%	69.8%	94.6%	76.8%
λέξεις συχνότητα >= 2	92.2%	58.2%	69.5%	95.1%	76.6%
λέξεις (stems) Porter stemmer	93.6%	49.6%	65.7%	96.6%	73.1%
λέξεις + bigrams	91.9%	73.0%	77.6%	93.6%	83.3%
λέξεις + bigrams συχνότητα >= 2	90.6%	74.5%	78.4%	92.3%	**83.4%**
λέξεις + bigrams (stems) Porter stemmer	91.3%	72.5%	77.2%	93.1%	82.8%

Πίνακας 4.4: Αποτελέσματα παραλλαγών Naive Bayes—Σύνολο κριτικών ταινιών

Τα ποσοστά ορθότητας κυμαίνονται στο 76.8% όταν χρησιμοποιείται αναπαράσταση bag-of-words με χρήση σκέτων λέξεων, και 83.4% όταν τα κείμενα αναπαρίστανται σαν bag-of-words με χρήση μεμονωμένων λέξεων και bigrams. Παρατηρούμε ότι, καταρχάς, τα ποσοστά της ορθότητας είναι ικανοποιητικά, αλλά όχι τόσο υψηλά όσο στο προηγούμενο dataset, δηλαδή πράγματι τα χαρακτηριστικά του συνόλου δεδομένων επηρεάζουν πολύ τα αποτελέσματα, και κατά μια έννοια το εύρος των αποτελεσμάτων ορθότητας που παρόμοια μοντέλα μπορούν να πετύχουν είναι και εγγενές στοιχείο του ίδιου του dataset.

Επιπροσθέτως, σε αυτό το σύνολο δεδομένων φαίνεται η χρήση των bigrams να δίνει μεγάλη βοήθεια στο μοντέλο, αυξάνοντας την ορθότητα κατά 6.6 ποσοστιαίες μονάδες. Αντιθέτως, το αν οι λέξεις με μια μόνο εμφάνιση αγνοηθούν και πάλι επηρεάζει λίγο το αποτέλεσμα. Επίσης, η αντικατάσταση λέξεων από τη ρίζα τους (stem) μειώνει την ορθότητα. Αυτό πιθανόν να οφείλεται στο ότι το stemming είναι πιο δύσκολο στην αγγλική γλώσσα απ' ότι στα ελληνικά, οπότε ο αλγόριθμος Porter μοιραία κάνει λάθη (π.χ. θεωρεί ότι το όνομα "James" έχει ρίζα το "Jame", σαν να ήταν πληθυντικός). Άλλος λόγος ίσως είναι το ότι στα αγγλικά υπάρχει μικρή διαφοροποίηση στις καταλήξεις σε σχέση με τα ελληνικά (δεν υπάρχουν π.χ. πρόσωπα ή πτώσεις), οπότε υπάρχει ούτως ή άλλως μικρό περιθώριο να κανονικοποιηθούν διαφορετικοί λεκτικοί τύποι στο ίδιο χαρακτηριστικό.

Μια επιπλέον παρατήρηση είναι ότι το μοντέλο μας εμφανίζει προκατάληψη προς τις αρνητικές κριτικές. Το γεγονός ότι οι αρνητικές κριτικές έχουν μεγάλο recall αλλά μικρό precision, και οι θετικές κριτικές το αντίστροφο, ση-

μαίνει ότι το μοντέλο αποφαίνεται συχνότερα ότι μια κριτική είναι αρνητική. Η ανισορροπία αυτή είναι μικρότερη όταν χρησιμοποιούνται και τα bigrams στα χαρακτηριστικά των κειμένων. Αφού όμως στο καλύτερο μοντέλο μας το precision των αρνητικών είναι 78.4% και το recall των θετικών 74.5%, αυτά τα νούμερα δεν είναι τόσο χαμηλά ώστε το μοντέλο να καταντάει άχρηστο.

Όπως και για το σύνολο δεδομένων κριτικών ξενοδοχείων, η επισκόπηση των πιο θετικών και πιο αρνητικών λέξεων μας παρέχει περισσότερη πληροφορία για το τι μαθαίνει το μοντέλο. Όταν το μοντέλο εκπαιδεύεται σε ολόκληρο το σύνολο δεδομένων, οι πιο χαρακτηριστικές μεμονωμένες λέξεις φαίνονται στον πίνακα 4.5 και τα πιο χαρακτηριστικά bigrams στον πίνακα 4.6.

Παρατηρούμε και πάλι πολλές λέξεις (και bigrams) τα οποία πολύ λογικά χαρακτηρίζουν θετικές ή αρνητικές κριτικές. Αλλά όμως παρατηρούμε και overfitting. Το πιο χαρακτηριστικό είναι η εμφάνιση ονομάτων συγκεκριμένων ταινιών ή συντελεστών να θεωρείται δείγμα θετικής ή αρνητικής γνώμης (π.χ. τα χαρακτηριστικά "meryl", "director james" και "tobey maguire" εμφανίζονται ανάμεσα στα πιο θετικά, ενώ τα "croft" και "forsythe" στα αρνητικά), κάτι που πιθανόν να μη γενικεύεται καλά, εκτός και αν ο εν λόγω συντελεστής συμμετέχει σε ταινίες που αρέσουν πάντα σε όλους ή ποτέ σε κανένα.

4.4 Lexicalized Hidden Markov Model Integrating Part-of-Speech

4.4.1 Τρόπος μέτρησης της αποτελεσματικότητας

Η απόδοση του μοντέλου Lexicalized HMM Integrating POS μετρήθηκε επίσης βάσει της ακρίβειας (precision) και της ανάκλησης (recall). Όμως καθώς η έξοδος του μοντέλου συνίσταται στην επισήμανση λέξεων με ετικέτες, οι δύο αυτές μετρικές ορίστηκαν διαφορετικά.

Καθώς η έξοδος του μοντέλου αυτού δεν είναι μια συνολική εκτίμηση για όλο το κείμενο, αλλά εντοπισμός συγκεκριμένων λέξεων-φράσεων που δηλώνουν θετική/αρνητική γνώμη αλλά και εντοπισμός λέξεων-φράσεων που αναφέρονται σε πλευρές των ξενοδοχείων, το precision και το recall ορίστηκαν ως εξής:

- Precision: Από όσες ετικέτες επισημάνθηκαν από το μοντέλο, το ποσοστό αυτών που ταίριαζαν με τις επισημάνσεις των ταξινομημένων δεδομένων.
- Recall: Από όλες τις επισημάνσεις στα ταξινομημένα δεδομένα, το ποσοστό αυτών που εντοπίστηκαν σωστά από το μοντέλο.

Σαν επισημάνσεις όμως δε μετριούνται οι ετικέτες BG, δηλαδή οι αδιάφορες λέξεις. Για παράδειγμα, στην φράση από τα δεδομένα μας "Η συμπεριφορα

Οι πιο αρνητικές λέξεις	
Λέξη	Πιθανότητα θετικού σχολίου
degenerates	0.0610
horrid	0.0765
pathetically	0.0765
tedium	0.0765
chevy	0.0835
leaden	0.0835
plodding	0.0835
ludicrous	0.0849
insulting	0.0885
sucks	0.0914
hodgepodge	0.0921
hyams	0.0921
setups	0.0921
stalks	0.0921
undeveloped	0.0921
3000	0.0965
stupidity	0.1017
hudson	0.1019
artemus	0.1025
batgirl	0.1025
campiness	0.1025
cinemax	0.1025
consecutive	0.1025
croft	0.1025
forsythe	0.1025
hmmm	0.1025
macdonald	0.1025
popped	0.1025
pseudonym	0.1025
rash	0.1025

Οι πιο θετικές λέξεις	
Λέξη	Πιθανότητα θετικού σχολίου
en	0.9375
lovingly	0.9375
melancholy	0.9333
gattaca	0.9286
ideals	0.9286
missteps	0.9286
masterfully	0.9231
burbank	0.9167
comforts	0.9167
criticized	0.9167
ideology	0.9167
meryl	0.9167
tobey	0.9167
brisk	0.9091
crimson	0.9091
downside	0.9091
envy	0.9091
exhilarating	0.9091
freed	0.9091
hypocrisy	0.9091
lithgow	0.9091
methodical	0.9091
niccol	0.9091
notoriety	0.9091
online	0.9091
soviet	0.9091
sullivan	0.9091
uncut	0.9091
outstanding	0.9073
avoids	0.9047

Πίνακας 4.5: Πιο ενδεικτικές λέξεις—κριτικές ταινιών

Τα πιο αρνητικά bigrams	
Bigram	Πιθανότητα θετικού σχολίου
this mess	0.0333
this turkey	0.0391
worst movie	0.0429
falls flat	0.0501
just bad	0.0697
really matter	0.0697
waste your	0.0697
worst movies	0.0697
and boring	0.0753
and laughable	0.0756
attempts humor	0.0756
degenerates into	0.0756
nothing much	0.0756
police academy	0.0756
reason care	0.0756
tries make	0.0756
trouble when	0.0756
wild wild	0.0756
not funny	0.0796
action scene	0.0826
are wasted	0.0826
chevy chase	0.0826
even care	0.0826
hard earned	0.0826
haunted house	0.0826
having her	0.0826
insult injury	0.0826
poison ivy	0.0826
randy quaid	0.0826
rent the	0.0826

Τα πιο θετικά bigrams	
Bigram	Πιθανότητα θετικού σχολίου
characters with	0.9474
very effective	0.9375
emotional and	0.9333
fits the	0.9286
american history	0.9231
director james	0.9231
film great	0.9231
intense and	0.9231
right time	0.9231
sharp and	0.9231
that deserves	0.9231
that once	0.9231
and moving	0.9167
many critics	0.9167
often hilarious	0.9167
pure entertainment	0.9167
the normal	0.9167
the strongest	0.9167
tobey maguire	0.9167
truman burbank	0.9167
are easily	0.9091
army and	0.9091
between these	0.9091
day day	0.9091
everyday life	0.9091
john lithgow	0.9091
lost her	0.9091
presents the	0.9091
the italian	0.9091
the mpaa	0.9091

Πίνακας 4.6: Πιο ενδεικτικά bigrams—κριτικές ταινιών

και η εξυπηρετηση του προσωπικου ηταν αριστη." η σωστή ανάθεση ετικετών (που έγινε από εμάς) έγινε σε 4 λέξεις και ήταν:

{ συμπεριφορα:SERVICE, εξυπηρετηση:SERVICE, προσωπικου:STAFF, αριστη:POSITIVE }

και σε όλες τις άλλες λέξεις ανατέθηκε η ετικέτα BG (αδιάφορες). Για να έχει recall 100% σε αυτό το κείμενο το μοντέλο θα πρέπει να επισημάνει και τις 4 αυτές λέξεις με τη σωστή ετικέτα. Το precision θα καθοριστεί από την ορθότητα όσων μη αδιάφορων ετικετών επισημανθούν.

Μια εκδοχή του μοντέλου τοποθέτησε τις ετικέτες:

{ συμπεριφορα:SERVICE, εξυπηρετηση:SERVICE, αριστη:NEGATIVE }

και σε όλες τις άλλες λέξεις ανατέθηκε η ετικέτα BG (αδιάφορες). Για το precision, παρατηρούμε ότι οι 2 από τις 3 ετικέτες είναι σωστές, δηλαδή το precision είναι 67%. Για το recall παρατηρούμε ότι 2 από τις 4 ετικέτες επισημάνθηκαν σωστά. Από τις δύο που το μοντέλο δεν πέτυχε, τη μία την επισήμανε σε λάθος κατηγορία (αριστη:NEGATIVE) και την άλλη καθόλου (προσωπικου:BG). Έτσι το recall ήταν 50%.

Τέλος, για να καταλήγουμε και σε μια μετρική η οποία αποτυπώνεται σε ένα νούμερο, χρησιμοποιούμε και το F-score που προκύπτει από το precision και το recall.

4.4.2 Δοκιμή σε κριτικές ξενοδοχείων

Η μέθοδος αυτή απαιτεί από τη φύση της διαφορετικό είδος ταξινομημένων δεδομένων από ότι η προηγούμενη. Ενώ το μοντέλο Naive Bayes (και στη συνέχεια όπως θα δούμε το Νευρωνικό Δίκτυο) αποφαίνεται για ένα κείμενο στο σύνολό του αν είναι θετικό ή αρνητικό, η μέθοδος που χρησιμοποιεί το κρυφό μοντέλο Markov εντοπίζει εντός του κειμένου συγκεκριμένες λέξεις ή φράσεις που αφορούν είτε έκφραση αρνητικής/θετικής άποψης είτε εκφάνσεις του υπό κριτική προϊόντος (εν προκειμένω το ξενοδοχείο). Για να μπορεί να μάθει να αναγνωρίζει αυτές τις λέξεις/φράσεις χρειάζεται επισήμανση στα κείμενα εκπαίδευσης πέρα από την ταξινόμησή τους ως συνολικά αρνητικά ή θετικά (που υπήρχε έτοιμη στο site όπου βρέθηκαν), αλλά με επισημειώσεις σε επίπεδο λέξεων.

Έτσι, ένα υποσύνολο από το dataset των κριτικών ξενοδοχείων σημειώθηκε χειρονακτικά με συγκεκριμένες ετικέτες σε συγκεκριμένες λέξεις από εμάς. Έτσι, ανατέθηκαν ετικέτες σε επίπεδο λέξης σε 196 θετικά κείμενα και 273 αρνητικά κείμενα, συνολικά 469 κείμενα. Το πόσες φορές ανατέθηκε η κάθε ετικέτα φαίνεται στον πίνακα 4.7. Από τον πίνακα φαίνεται το γιατί σημειώσαμε με ετικέτες περισσότερα αρνητικά από ότι θετικά κείμενα. Ο λόγος είναι ότι στα αρνητικά κείμενα οι χρήστες χρησιμοποιούσαν λιγότερες αρνητικές λέξεις, όπως π.χ. το επίθετο "κακός", ενδεχομένως από ευγένεια. Για να μάθει όμως το μοντέλο αυτό και αρνητικές λέξεις χρειάστηκε να βάλουμε ετικέτες σε περισσότερα αρνητικά κείμενα, ώστε ο αριθμός των σημειωμένων αρνητικών

λέξεων να πλησιάσει αυτόν των θετικών λέξεων.

Ετικέτα	Αριθμός αναθέσεων
POSITIVE	806
NEGATIVE	514
SERVICE	509
BUILDING	436
LOCATION	235
STAFF	169
COST	90
Σύνολο	2759

Πίνακας 4.7: Συχνότητες αναθέσεων ετικετών στο training set

Στη δημοσίευση με βάση την οποία υλοποιήσαμε αυτό το σύστημα [JHS09], οι συγγραφείς δημιούργησαν training set από κριτικές για κάποια μοντέλα ψηφιακών καμερών, και test set από κριτικές για κάποια διαφορετικά μοντέλα, ούτως ώστε να μην επαναλαμβάνονται οι ίδιες ακριβώς εκφάνσεις (aspects) του προϊόντος. Ακολουθώντας αυτή τη λογική, αντί να εκπαιδεύουμε και να ελέγχουμε από το ίδιο dataset κριτικών ξενοδοχείων της Αττικής, συλλέξαμε ένα δεύτερο, μικρότερο dataset με κριτικές ξενοδοχείων από το νησί της Τήνου (όλες όσες ήταν διαθέσιμες τη στιγμή της συλλογής από το site). Αυτό το σύνολο αποτελούνταν από 146 θετικά σχόλια και 67 αρνητικά σχόλια, συνολικά 213 κείμενα, και έγιναν όλα tagged χειροκίνητα, ως προς τις εκφάνσεις των ξενοδοχείων και την έκφραση αρνητικής και θετικής γνώμης, με ίδια διαδικασία όπως και το training set. Το πλήθος των αναθέσεων των διαφόρων ετικετών στο test set φαίνεται στον πίνακα 4.8.

POSITIVE	468
SERVICE	219
LOCATION	124
BUILDING	100
STAFF	90
NEGATIVE	77
COST	26
Σύνολο	1104

Πίνακας 4.8: Συχνότητες αναθέσεων ετικετών στο test set

Η μέθοδος ελέγχου εδώ συνίσταται στην εκπαίδευση στο training set και στην επαλήθευση στο test set, όπου συγκρίνονται οι ετικέτες που εισάγαμε χειρονακτικά με αυτές που τοποθέτησε το μοντέλο. Η απόδοση ελέγχθηκε βά-

σει των precision και recall στο σύνολο των ετικετών, και στο συνδυασμένο F-score, όπως αναφέρθηκε παραπάνω.

Διερεύνηση παραμέτρων μοντέλου

Καθώς το μοντέλο Lexicalized HMM Integrating Part-of-Speech μπορεί να παραμετροποιηθεί ως προς κάποιες απόψεις, εκτελέσαμε δοκιμές για διαφορετικές τιμές των ποσοτικών παραμέτρων και επιλογές των ποιοτικών παραμέτρων και αποφανθήκαμε για το ποιες δουλεύουν καλύτερα.

Οι ποσοτικές παράμετροι του μοντέλου είναι οι συντελεστές γραμμικής εξομάλυνσης μεταξύ των lexicalized και non-lexicalized εκδοχών του μοντέλου, δηλαδή το κατά πόσο η εμφάνιση της επόμενης ετικέτας, μέρους του λόγου ή λέξης εξαρτάται από την προηγούμενη λέξη.

Οι ποιοτικές παράμετροι είναι η επιλογή του σημειωτή μέρους του λόγου (part-of-speech tagger). Εδώ δοκιμάστηκαν τέσσερις επιλογές:

- Σημειωτής μέρους του λόγου του Ινστιτούτου Επεξεργασίας του Λόγου: `ILSP_POS_TAGGER`.
- Ο δικός μας απλούστερος σημειωτής μέρους του λόγου: `MY_POS_TAGGER`.
- Χρήση της κατάληξης της λέξης αντί μέρους του λόγου, όπως αυτή αποκόπτεται από τον stemmer: `SUFFIX_TAGGER`. Η ιδέα είναι ότι οι καταλήξεις των λέξεων σχετίζονται με το μέρος του λόγου τους (π.χ. -ήκαμε είναι κατάληξη μόνο ρήματος), οπότε αν αυτή η αντικατάσταση δούλευε, θα αποφεύγαμε την ανάγκη για πραγματικό σημειωτή μέρους του λόγου.
- Μη χρήση σημειωτή μέρους του λόγου. Χρησιμοποιήθηκε μια dummy κλάση η οποία επέστρεφε αυτόματα το ίδιο "μέρος του λόγου" (NONE) για κάθε λέξη: `NO_POS_TAGGER`. Το αποτέλεσμα είναι να μηδενίζεται η συνεισφορά του σημειωτή μέρους του λόγου, ώστε να δούμε πως δουλεύει το μοντέλο στην περίπτωση αυτή.

Άλλη μια ποιοτική παράμετρος είναι εάν στη θέση των λέξεων χρησιμοποιούνται τα stems αυτών. Έγιναν δοκιμές με και χωρίς αυτήν την προσθήκη. Τέλος, ακόμα και αν δεν αντικαθίστανται οι λέξεις από τα stems, γίνεται πάντα κανονικοποίηση ως προς τους τόνους και τα κεφαλαία, καθώς αυτή πάντα βοηθάει στην απόδοση.

Πρώτα δοκιμάσαμε να μετρήσουμε πως κινείται η απόδοση σε σχέση με τους συντελεστές γραμμικής εξομάλυνσης `word_coeff`, `tag_coeff` και `pos_coeff`. Αυτοί οι τρεις σε πρώτη φάση θέτονταν στην ίδια τιμή `coeff` για να δούμε την επιρροή της εξομάλυνσης συνολικά. Έτσι, εκτελέσαμε το πείραμα (εκπαιδεύοντας στο training set και ελέγχοντας με το test set) για τις ακόλουθες τιμές του κοινού συντελεστή `coeff`:

$$\{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.995, 0.999, 1.0\}$$

.

Οι τιμές μεταξύ 0.9 και 1.0 δοκιμάστηκαν γιατί παρατηρήθηκε αύξηση της απόδοσης σε κάποια μοντέλα σε αυτό το εύρος τιμών. Όλες οι τιμές δοκιμάστηκαν χρησιμοποιώντας και τους 4 σημειωτές μέρους του λόγου, οπότε έγιναν συνολικά $15 \times 4 = 60$ πειράματα.

Στον πίνακα 4.9 κάθε γραμμή αντιστοιχεί σε μια τιμή για τον κοινό συντελεστή `coeff`. Κάθε τρεις στήλες εμφανίζονται οι τιμές Precision, Recall και F-Score (P, R και F αντίστοιχα) που προέκυψαν χρησιμοποιώντας το σημειωτή μέρους του λόγου που φαίνεται πιο ψηλά στον πίνακα.

Από τα αποτελέσματα του πίνακα 4.9 προκύπτει ότι:

- Τα επίπεδα της ανάκλησης είναι γενικώς χαμηλά. Στις περισσότερες περιπτώσεις είναι περίπου στο 50%. Αυτό σημαίνει ότι οι μισές επισημάνσεις που έγιναν στο test set δεν εντοπίστηκαν από το μοντέλο, κάτι που επιβεβαιώσαμε κατά την ανάπτυξη του συστήματος παρατηρώντας τις εξόδους του. Πολλές λέξεις, είτε αφορούν σε οντότητες εκφάνσεων των ξενοδοχείων, είτε οντότητες έκφρασης άποψης, δε σημειώνονται καθόλου (συνήθως λαμβάνουν την ετικέτα BG που αντιστοιχεί σε αδιάφορη λέξη).

- Η ακρίβεια αντίθετα είναι πιο ικανοποιητική. Οι ετικέτες που αναθέτονται είναι σωστές σε ποσοστά περίπου 75%-80% στα τρία πρώτα μοντέλα και 85%-90% στο μοντέλο χωρίς POS tagger.

- Ο σχολιασμός της ανάκλησης και της ακρίβειας δε μπορεί να γίνει όμως χωριστά. Σε αυτά τα αποτελέσματα είναι προφανής η αντίστροφη σχέση των δύο.

- Το μοντέλο χωρίς POS tagger για παράδειγμα έχει με διαφορά τη μεγαλύτερη ακρίβεια αλλά και τη μικρότερη ανάκληση, πρόκειται δηλαδή για συντηρητικό μοντέλο που κάνει μόνο τις πιο σίγουρες επιλογές. Μια πιθανή εξήγηση για αυτή τη συμπεριφορά είναι ότι χωρίς μέρος του λόγου το μοντέλο δεν διευκολύνεται να "μαντέψει" την ετικέτα που αντιστοιχεί σε άγνωστες λέξεις, αλλά αναθέτει ετικέτες μόνο όταν η ίδια ακριβώς λέξη έχει σημειωθεί με την ίδια ετικέτα στα δεδομένα εκπαίδευσης. Αυτό το παρατηρήσαμε με συνηθισμένες λέξεις όπως "καλό" για θετική άποψη, "πρωινό" για την έκφανση SERVICE και "δωμάτιο" για την έκφανση BUILDING.

- Χρησιμοποιώντας POS tagger (ή υποκατάστατο αυτού στην περίπτωση του `SUFFIX_TAGGER`) παρατηρούμε αύξηση της ανάκλησης και μείωση της ακρίβειας. Τα μοντέλα αναθέτουν περισσότερες ετικέτες αλλά κάνουν και περισσότερα λάθη.

- Συγκρίνοντας βάσει F-Score για τη βέλτιστη τιμή `coeff` του κάθε μοντέλου, οι σημειωτές μέρους του λόγου ακολουθούν την ακόλουθη κατάταξη:

tagger / coeff	ILSP POS TAGGER			MY POS TAGGER		
	P	R	F	P	R	F
0.0	74.0%	47.6%	57.9%	71.5%	49.5%	58.5%
0.1	75.7%	46.5%	57.6%	73.5%	50.1%	59.6%
0.2	76.9%	46.5%	57.9%	74.4%	50.6%	60.3%
0.3	77.9%	46.2%	58.0%	74.8%	51.0%	60.6%
0.4	78.0%	45.9%	57.8%	75.9%	50.3%	60.5%
0.5	78.9%	46.5%	**58.5%**	76.9%	50.2%	60.7%
0.6	78.6%	45.9%	58.0%	78.2%	49.9%	60.9%
0.7	78.4%	44.9%	57.1%	79.5%	50.5%	61.7%
0.8	78.0%	44.3%	56.5%	79.3%	50.2%	61.5%
0.9	77.2%	43.8%	55.9%	79.5%	50.0%	61.4%
0.95	77.0%	44.4%	56.3%	78.9%	51.4%	62.2%
0.99	77.3%	45.7%	57.4%	78.1%	52.4%	**62.8%**
0.995	77.8%	46.0%	57.8%	77.5%	52.7%	62.7%
0.999	77.6%	46.0%	57.8%	76.2%	52.7%	62.3%
1.0	66.4%	46.4%	54.6%	67.1%	53.0%	59.2%
tagger / coeff	SUFFIX TAGGER			NO POS TAGGER		
	P	R	F	P	R	F
0.0	79.4%	47.1%	59.1%	86.0%	37.2%	52.0%
0.1	80.5%	46.4%	58.9%	89.5%	36.3%	51.7%
0.2	80.8%	46.4%	58.9%	90.9%	35.1%	50.7%
0.3	80.4%	46.0%	58.5%	91.2%	34.6%	50.2%
0.4	79.3%	46.0%	58.2%	90.9%	35.2%	50.8%
0.5	78.6%	45.9%	58.0%	90.1%	34.6%	50.0%
0.6	79.1%	46.2%	58.3%	89.6%	34.2%	49.5%
0.7	78.3%	46.4%	58.2%	89.1%	35.6%	50.9%
0.8	77.4%	46.3%	57.9%	87.7%	36.7%	51.7%
0.9	76.0%	46.3%	57.5%	88.1%	39.4%	54.4%
0.95	76.1%	47.1%	58.2%	86.8%	40.5%	55.2%
0.99	76.7%	48.8%	**59.7%**	85.6%	43.6%	57.7%
0.995	76.0%	48.6%	59.3%	85.5%	43.8%	58.0%
0.999	75.5%	48.9%	59.4%	84.7%	44.5%	**58.3%**
1.0	65.9%	48.9%	56.1%	69.4%	44.9%	54.5%

Πίνακας 4.9: Αποτελέσματα Hidden Markov Model: Επίδραση σημειωτή μέρους του λόγου και συντελεστή γραμμικής εξομάλυνσης

1. MY POS TAGGER: 62.8%
2. SUFFIX TAGGER: 59.7%
3. ILSP POS TAGGER: 58.5%
4. NO POS TAGGER: 58.3%

- Είναι ενδιαφέρον το γεγονός ότι ένας "πραγματικός" σημειωτής μέρους του λόγου αποδεικνύεται λιγότερο βοηθητικός για το μοντέλο μας από τον πολύ απλούστερο δικό μας. Ενδεχομένως όμως το γεγονός ότι ο "MY POS TAGGER" κατατάσσει τις λέξεις σε μια από 12 κατηγορίες (βλέπε πίνακα 3.1) αντί για τις εκατοντάδες του "ILSP POS TAGGER" να διευκολύνει το μοντέλο να μάθει τις σχέσεις μεταξύ μέρους του λόγου και ετικετών. Ίσως ο "ILSP POS TAGGER" να απαιτεί έτσι μεγαλύτερο σύνολο εκπαίδευσης για να συνδράμει πιο θετικά.

- Δεν μπορούμε να αποφανθούμε ξεκάθαρα για την ιδανική τιμή του κοινού συντελεστή εξομάλυνσης μεταξύ των lexicalized και των non-lexicalized πιθανοτήτων. Το μόνο ξεκάθαρο συμπέρασμα είναι πως μόλις αυτός γίνει ακριβώς 1.0, δηλαδή όταν το απλούστερο non-lexicalized μοντέλο δε χρησιμοποιείται, η ακρίβεια πέφτει απότομα χωρίς να αυξάνεται η ανάκληση και έτσι μειώνεται το F-Score. Για καλύτερη ανάλυση, χρειάστηκαν πειράματα όπου οι τρεις συντελεστές λάμβαναν ξεχωριστές τιμές.

Στη συνέχεια, κάναμε δοκιμές όπου οι τρεις συντελεστές εξομάλυνσης `word_coeff`, `tag_coeff` και `pos_coeff` λάμβαναν διαφορετικές τιμές. Επιλέγοντας τιμές γι αυτούς από το σύνολο $\{0.0, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$, εκτελέσαμε το πείραμα για όλους τους $7^3 = 343$ διαφορετικούς συνδυασμούς. Αυτό το πείραμα έγινε τρεις φορές:

- Με χρήση του δικού μας σημειωτή `MY_POS_TAGGER`.

- Με χρήση του σημειωτή του ILSP `ILSP_POS_TAGGER`.

- Με χρήση του δικού μας σημειωτή `MY_POS_TAGGER` και ύστερα αντικατάσταση των λέξεων από το stem της.

Στους παρακάτω πίνακες (4.10, 4.11 και 4.12) δείχνουμε τα καλύτερα 20 μοντέλα για καθεμιά από τις τρεις κατηγορίες, ταξινομημένα με βάση το F-Score (τελευταία στήλη).

Συμπεράσματα:

- Καταρχάς εδώ συναντάμε το καλύτερο ως τώρα μοντέλο, με χρήση του "MY POS TAGGER" και συντελεστές εξομάλυνσης `word_coeff` $= 0.1$, `tag_coeff` $= 0.9$ και `pos_coeff` $= 0.9$, με το υψηλότερο F-Score (65.0%), ακρίβεια 79.8% και ανάκληση 54.8%.

word_coeff	tag_coeff	pos_coeff	P	R	F
0.0	0.7	0.5	72.7%	54.3%	62.2%
0.7	0.9	0.7	71.9%	54.8%	62.2%
0.9	1.0	0.9	70.5%	55.8%	62.3%
0.5	1.0	0.9	68.1%	57.7%	62.5%
0.7	1.0	0.9	69.7%	56.8%	62.6%
0.5	0.9	1.0	81.3%	51.1%	62.7%
0.0	0.7	0.7	78.2%	52.6%	62.9%
0.1	0.7	0.7	78.4%	52.5%	62.9%
0.1	1.0	1.0	70.7%	56.7%	62.9%
0.5	1.0	1.0	72.0%	56.1%	63.0%
0.0	1.0	1.0	70.8%	57.1%	63.2%
0.3	1.0	1.0	71.9%	56.3%	63.2%
0.7	0.9	0.9	80.0%	52.8%	63.6%
0.3	0.9	1.0	82.0%	52.8%	64.2%
0.5	0.9	0.9	79.9%	53.8%	64.3%
0.0	0.9	1.0	81.5%	53.5%	64.6%
0.1	0.9	1.0	81.7%	53.4%	64.6%
0.3	0.9	0.9	79.9%	54.2%	64.6%
0.0	0.9	0.9	79.3%	54.8%	64.8%
0.1	0.9	0.9	79.8%	54.8%	65.0%

Πίνακας 4.10: Αποτελέσματα Hidden Markov Model: Καλύτερες 20 επιλογές συντελεστών γραμμικής εξομάλυνσης, σημειωτής MY_POS_TAGGER

word_coeff	tag_coeff	pos_coeff	P	R	F
0.0	0.7	0.5	77.7%	50.5%	61.3%
0.0	0.9	0.9	80.9%	49.4%	61.3%
0.1	1.0	0.5	65.9%	57.2%	61.3%
0.5	0.9	0.5	71.0%	54.0%	61.3%
0.0	1.0	0.3	64.0%	59.0%	61.4%
0.1	1.0	0.7	67.8%	56.3%	61.5%
0.3	1.0	0.9	71.5%	53.9%	61.5%
0.0	1.0	0.7	68.0%	56.5%	61.8%
0.3	0.9	0.3	68.8%	56.1%	61.8%
0.1	1.0	0.9	71.9%	54.3%	61.9%
0.3	0.9	0.5	71.4%	54.7%	61.9%
0.3	0.9	0.7	76.4%	52.1%	61.9%
0.0	1.0	0.5	66.7%	57.9%	62.0%
0.0	1.0	0.9	71.7%	54.6%	62.0%
0.1	0.9	0.3	69.0%	56.2%	62.0%
0.1	0.9	0.5	71.8%	54.9%	62.2%
0.0	0.9	0.3	69.5%	56.4%	62.3%
0.1	0.9	0.7	76.7%	52.5%	62.4%
0.0	0.9	0.5	72.1%	55.2%	62.5%
0.0	0.9	0.7	77.0%	52.8%	62.7%

Πίνακας 4.11: Αποτελέσματα Hidden Markov Model: Καλύτερες 20 επιλογές συντελεστών γραμμικής εξομάλυνσης, σημειωτής ILSP_POS_TAGGER

word_coeff	tag_coeff	pos_coeff	P	R	F
0.1	0.7	0.7	78.5%	51.4%	62.1%
0.0	0.7	0.7	78.6%	51.9%	62.5%
0.5	0.9	0.9	79.0%	51.7%	62.5%
0.3	1.0	0.9	69.4%	57.0%	62.6%
0.0	0.9	0.9	75.4%	53.6%	62.7%
0.1	1.0	0.9	69.5%	57.2%	62.8%
0.0	1.0	0.9	69.4%	57.4%	62.9%
0.0	1.0	1.0	70.9%	56.6%	62.9%
0.5	0.9	1.0	82.6%	50.9%	63.0%
0.1	0.9	0.9	77.5%	53.3%	63.1%
0.1	1.0	1.0	71.3%	56.5%	63.1%
0.3	0.9	0.9	79.4%	52.4%	63.1%
0.7	1.0	1.0	74.5%	54.9%	63.2%
0.5	1.0	0.9	72.1%	56.6%	63.4%
0.7	1.0	0.9	73.1%	56.1%	63.5%
0.3	0.9	1.0	82.4%	51.7%	63.6%
0.3	1.0	1.0	73.9%	56.3%	63.9%
0.5	1.0	1.0	75.1%	55.9%	64.1%
0.1	0.9	1.0	82.6%	52.6%	64.3%
0.0	0.9	1.0	81.7%	53.4%	64.6%

Πίνακας 4.12: Αποτελέσματα Hidden Markov Model: Καλύτερες 20 επιλογές συντελεστών γραμμικής εξομάλυνσης, σημειωτής `MY_POS_TAGGER`, χρήση stems αντί για λέξεις

- Επίσης, η χρήση stems αντί για λέξεις δε φαίνεται να βελτιώνει την απόδοση του μοντέλου, χωρίς να τη βλάπτει όμως, με υψηλότερο F-Score να είναι 64.6%, για συντελεστές `word_coeff` $= 0.0$, `tag_coeff` $= 0.9$ και `pos_coeff` $= 1.0$.

- Φαίνεται ότι ο συντελεστής `word_coeff`, δηλαδή το βάρος που δίνεται στην κατανομή πιθανότητας όπου η εμφάνιση μιας λέξης εξαρτάται και από την προηγούμενη λέξη, φαίνεται να είναι καλύτερο να έχει μικρή ή και μηδενική τιμή. Δηλαδή αυτή η εξάρτηση συνεισφέρει λιγότερο στο να βελτιώνεται το αποτέλεσμα.

- Ο συντελεστής που έχει πιο εμφανή βέλτιστη τιμή (ανάμεσα σε αυτές που δοκιμάστηκαν) είναι ο `tag_coeff` και η τιμή αυτή είναι 0.9 . Αυτό δείχνει ότι το να εξαρτάται η τρέχουσα ετικέτα από την προηγούμενη ετικέτα και την προηγούμενη λέξη είναι μια καλή ιδέα για το μοντέλο. Πράγματι, στα δεδομένα εκπαίδευσης και ελέγχου συναντώνται συχνά μοτίβα συσχετισμού λέξης και επόμενης ετικέτας. Για παράδειγμα, οι λέξεις "πολύ" και "ήταν" ακολουθούνται συχνά από λέξη με ετικέτα POSITIVE ή NEGATIVE.

- Ο συντελεστής `pos_coeff` είναι προτιμότερο να έχει τιμή 0.9 όταν χρησιμοποιείται ο "MY POS TAGGER", ενώ φαίνεται να μην επηρεάζει πολύ όταν χρησιμοποιείται ο "ILSP POS TAGGER", πιθανόν επειδή με τα πολλά διαφορετικά μέρη του λόγου που αυτός αποδίδει, το μοντέλο δεν μπορεί να μάθει χρήσιμες συσχετίσεις.

4.4.3 Ανακάλυψη νέων λέξεων και φράσεων

Ένα ποιοτικό χαρακτηριστικό του μοντέλου αυτού, καθώς και των διεξαγόμενων πειραμάτων, είναι ότι πολλές από τις ετικέτες που αποδίδονται σωστά κατά τον έλεγχο αντιστοιχούν σε λέξεις οι οποίες συναντήθηκαν στο σύνολο εκπαίδευσης με την ίδια ακριβώς ετικέτα. Δηλαδή το μοντέλο έχει την τάση να εκφυλίζεται σε μια συμπεριφορά όπου το παραγόμενο αποτέλεσμα μοιάζει με απομνημόνευση συσχετίσεων λέξεων-ετικετών. Μάλιστα όσο μεγαλύτερο είναι το Precision μιας δοκιμής, τόσο περισσότερο συμβαίνει αυτό το φαινόμενο.

Σε μοντέλα όμως όπου πετυχαίνεται μεγαλύτερο Recall έχουμε παραδείγματα λέξεων ή φράσεων που δεν συναντήθηκαν καθόλου στο training set με μια δεδομένη ετικέτα, και όταν στο test set είχαν σημειωθεί με την ετικέτα αυτή (χειρονακτικά από εμάς), το μοντέλο τις αναγνώρισε σωστά, χωρίς να έχει απομνημονεύσει τη συσχέτιση της ίδιας της λέξης ή φράσης με την ετικέτα, αλλά αξιοποιώντας τις άλλες δυνατότητες του, δηλαδή τις συσχετίσεις μέρους του λόγου, προηγούμενης λέξης και προηγούμενης ετικέτας.

Μεμονωμένες "νέες" λέξεις που ανακάλυψε μια εκδοχή του μοντέλου, χωρίς να είχαν σημειωθεί έτσι στο training set, φαίνονται στον πίνακα 4.13:

Λέξη	Ετικέτα	Λέξη	Ετικέτα
ανεπανάληπτο	POSITIVE	αξιαγάπητος	POSITIVE
αξιόλογη	POSITIVE	απολαυστικό	POSITIVE
γραφικο	POSITIVE	εξαιρετικός	POSITIVE
εξοπλισμένο	POSITIVE	ευχαριστος	POSITIVE
εύγεστο	POSITIVE	ηρεμη	POSITIVE
θάλασσα	LOCATION	κακά	NEGATIVE
καλοσυνατοι	POSITIVE	μπαλκονακι	BUILDING
ξενοδοχος	STAFF	πανέμορφη	POSITIVE
πεντακάθαροι	POSITIVE	περιποιημένη	POSITIVE
πλήρης	POSITIVE	τραπεζάκι	SERVICE
φιλικοτατη	POSITIVE	χαμογελαστος	POSITIVE

Πίνακας 4.13: Άγνωστες λέξεις, σωστά σημειωμένες από το μοντέλο

Μια δυνατότητα του μοντέλου που δυστυχώς δεν αξιοποιήθηκε πολύ σε αυτό το σύνολο δεδομένων είναι η δυνατότητα επισήμανσης μικρών φράσεων με μια ετικέτα. Δε χρησιμοποιήθηκε πολύ, γιατί σε αντίθεση με το domain των ψηφιακών καμερών, οι εκφάνσεις του προϊόντος ξενοδοχείου συνήθως περιγράφονται με μια λέξη.

Μια περίπτωση που όμως αυτή η δυνατότητα δούλεψε καλά, είναι στην έκφανση της τοποθεσίας (LOCATION). Οι φράσεις που σημειώθηκαν στα δεδομένα εκπαίδευσης ανέφεραν ότι τη θέση του ξενοδοχείου σε σχέση με αξιοθέατα ή άλλα σημεία ενδιαφέροντος. Δείγμα φράσεων που σημειώθηκαν στα δεδομένα εκπαίδευσης ως LOCATION φαίνεται στον πίνακα 4.14.

Η εκμάθηση από το μοντέλο μοτίβων λέξεων και μερών του λόγου το βοήθησε να επισημάνει σωστά τις, προηγουμένως άγνωστες, φράσεις που φαίνονται στον πίνακα 4.15.

4.4.4 Απόδοση ανά ετικέτα

Όπως γράψαμε και στην υποενότητα 4.2.3, είναι σημαντικό να ελέγχουμε την ακρίβεια και την ανάκληση ανά κατηγορία ταξινόμησης για να διαπιστώνεται πιθανή προκατάληψη του μοντέλου προς κάποιες κατηγορίες. Έτσι, για το μοντέλο που πέτυχε το μεγαλύτερο F-Score σε όλες τις ετικέτες συνολικά, υπολογίσαμε τα Precision, Recall και F-Score για καθεμιά από τις 7 ετικέτες ξεχωριστά στον πίνακα 4.16.

Παρατηρούμε ότι η απόδοση δεν είναι ομοιόμορφη ανά ετικέτα. Συγκεκριμένα, από τις εκφάνσεις το STAFF έχει με διαφορά το καλύτερο precision, recall και F-score, κάτι που αποδίδεται στο ότι οι λέξεις που σημειώνονταν ως STAFF είχαν μικρότερη ποικιλία από ότι στις άλλες εκφάνσεις, κάνοντας πιο εύκολη την αναγνώρισή τους. Με μέτρο σύγκρισης το F-Score, ακολουθεί η

δίπλα απο σταθμό μ.	δίπλα στα μέσα μεταφορας	δίπλα στην αμμουδιά του Τολό
διπλα απο το μετρο	διπλα στο μετρο	δρόμος μπροστά από το ξενοδοχείο
κοντά με ταξί	κοντά σε μετρό και ταξί	κοντά σε στάση λεωφορείου
κοντά σε όλα τα ‘‘απαραίτητα‘‘	κοντά στην ομόνια	κοντά στην ομόνοια
κοντά στην παλιά πόλη	κοντά στο metro	κοντά στο κέντρο και τα μμε
κοντά στο κέντρο του Ναυπλίου	κοντά στο κέντρο του νησιού	κοντα στο μετρο
κοντά στο σταθμό του μετρό	κοντά στό κέντρο	κοντα σε μετρο
κοντα σε παραλια	κοντα σε σταθμο του μετρο	κοντα σε σταση του μετρο
κοντα στη θαλασσα	κοντα στην ομονοια	κοντα στο κεντρο και αυτο
κοντα στο κεντρο της γλυφαδας	λίγο μακριά από τη θάλασσα	μέσα στο κέντρο της πολης
μέσα στο κέντρο της πόλης	με θέα προς την ακρόπολη	μερος του νησιου
προσβαση στα εμπορικα καταστηματα	προσβαση στις συγκοινωνιες	προσβαση στο κεντρο και λιμανι
προσβαση στο κεντρο τησ αθηνασ	προσβαση στο μετρο	προσβασης σε υπηρεσιες με τα ποδια
πρόσβαση στο κεντρό της Αθήνας	πρόσβαση στο μετρό	σημείο της παραλίας
στάση μετρό στα 50 μέτρα	στην καρδιά της Αθήνας	στην καρδιά της παλιάς πόλης
στην πανεμορφη Πλακα	στην περιοχή της Γλυφάδας	στην πιο ήσυχη μεριά της πόλης
στο ιστορικο κεντρο του Ναυπλιου	στο κέντρο της Αθήνας	στο κέντρο της αθήνας

Πίνακας 4.14: Φράσεις σημειωμένες ως LOCATION στα δεδομένα εκπαίδευσης

Φράση	Ετικέτα
δίπλα στη θάλασσα	LOCATION
κοντά στην εκκλησία	LOCATION
κοντα στην πανεμορφη παραλια	LOCATION
κοντα στη χωρα	LOCATION
προσβαση στη πολη	LOCATION

Πίνακας 4.15: Άγνωστες φράσεις των δεδομένων ελέγχου, σωστά σημειωμένες από το μοντέλο

Ετικέτα	Precision	Recall	F-Score
SERVICE	79.4% (100/126)	45.7% (100/219)	58.0%
STAFF	90.8% (59/65)	65.6% (59/90)	76.1%
BUILDING	64.1% (66/103)	66.0% (66/100)	65.0%
LOCATION	73.2% (60/82)	48.4% (60/124)	58.3%
COST	68.4% (13/19)	50.0% (13/26)	57.8%
POSITIVE	86.1% (273/317)	58.3% (273/468)	69.6%
NEGATIVE	69.6% (39/56)	50.6% (39/77)	58.6%
Συνολικά	79.4% (610/768)	55.3% (610/1104)	65.2%

Πίνακας 4.16: Απόδοση ανά ετικέτα

έκφανση BUILDING και έπειτα οι LOCATION, SERVICE και COST, που έχουν σχεδόν ίσο F-Score. Γενικά υπάρχουν διαφορές στην απόδοση ανά ετικέτα έκφανσης, αλλά αυτό συμβαίνει γιατί οι 5 ετικέτες αντιστοιχούν σε διαφορετική ποικιλομορφία λέξεων και φράσεων. Π.χ. η έκφανση SERVICE έχει μεγάλο εύρος εφαρμογής στις διάφορες υπηρεσίες που προσφέρουν τα ξενοδοχεία (κάποιες από τις πιο ασυνήθιστες υπηρεσίες που επισημάναμε στο test set ήταν "πετσέτες παραλίας", "αφυγραντήρες" και "κλειδαριές", υπό την έννοια ότι δεν είχαν εμφανιστεί στα δεδομένα εκπαίδευσης και ήταν δύσκολο να επισημανθούν από το μοντέλο). Αντίθετα, στην έκφανση BUILDING που αφορά τις κτιριακές εγκαταστάσεις του ξενοδοχείου, οι λέξεις "δωμάτιο" και "κτήριο" είχαν σημειωθεί αρκετές φορές στα δεδομένα εκπαίδευσης και ήταν εύκολο να σημειωθούν σωστά από το μοντέλο στα δεδομένα ελέγχου.

Όσον αφορά τις ετικέτες έκφρασης άποψης, η ετικέτα POSITIVE έχει μαθευτεί καλύτερα από την ετικέτα NEGATIVE, με την πρώτη να υπερέχει σε Precision, Recall και F-Score κατά 16.5, 7.7 και 11 ποσοστιαίες μονάδες αντίστοιχα. Αυτό είναι πιθανό να συμβαίνει διότι στα δεδομένα εκπαίδευσης υπολείπονταν οι ετικέτες αρνητικής άποψης έναντι των θετικών. Έχοντας διαπιστώσει αυτή την ανισορροπία, επισημειώσαμε περισσότερα κείμενα αρνητικών στοιχείων, ώστε να αυξήσουμε τις αρνητικές ετικέτες στο training set (όπως αναφέραμε στην αρχή της υποενότητας 4.4.2). Έτσι, η αναγνώριση των

αρνητικών ετικετών βελτιώθηκε. Πιθανόν με επισημείωση ακόμα περισσότερων κειμένων (κυρίως αρνητικών) αυτή η ανισορροπία στον έλεγχο στο test set να μειωνόταν.

4.5 Νευρωνικά Δίκτυα

Καθώς το είδος των νευρωνικών δικτύων που χρησιμοποιήσαμε δεχόταν το κείμενο σαν bag-of-words, και το δίκτυο αποφαινόταν για το αν το κείμενο είναι συνολικά θετικό ή αρνητικό, οι παρακάτω δοκιμές είναι της ίδιας μορφής με αυτές του αλγορίθμου Naive Bayes. Αφού είναι συγκρίσιμες, στους πίνακες των αποτελεσμάτων παρατίθεται στο τέλος και η απόδοση του καλύτερου Naive Bayes μοντέλου ανά σύνολο δεδομένων, για να φανεί ποιο μοντέλο υπερτερεί.

4.5.1 Δοκιμή σε κριτικές ξενοδοχείων

Πραγματοποιήθηκαν πειράματα τύπου 10-fold cross-validation. Ο διαχωρισμός του κειμένου σε λέξεις έγινε όπως και προηγουμένως (βλέπε υποενότητα 4.3.1).

Για να γίνει σύγκριση και με ένα απλούστερο νευρωνικό δίκτυο, εκτός από το δίκτυο τριών επιπέδων, υλοποιήθηκε και δοκιμάστηκε και μια απλούστερη εκδοχή του, το perceptron ενός επιπέδου (single-layer perceptron). Αυτό δε διαθέτει κρυφό επίπεδο, παρά μόνο ένα επίπεδο από νευρώνες μεταξύ των εισόδων και των εξόδων.

Το single-layer perceptron έχει ως παράμετρο το ποια ήταν τα features του κειμένου που χρησιμοποιήθηκαν, καθώς και για πόσες εποχές (epochs) εκπαιδεύτηκε, δηλαδή για πόσα πλήρη περάσματα έγιναν από το σύνολο εκπαίδευσης.

Το three-layer δίκτυο παραμετροποιείται ως προς την επιλογή των features, το για πόσες εποχές θα εκπαιδευτεί αλλά και το πλήθος των νευρώνων στο κρυφό επίπεδο (hidden layer size).

Τα αποτελέσματα φαίνονται στον πίνακα 4.17, όπου στο τέλος υπάρχει και ο αλγόριθμος Naive Bayes για σύγκριση.

Παρατηρούμε ότι τα νευρωνικά δίκτυα που δοκιμάστηκαν δεν ξεπέρασαν την απόδοση του απλούστερου Naive Bayes. Ο λόγος που συμβαίνει αυτό είναι πιθανότερα ότι πρόκειται για εφαρμογή νευρωνικών δικτύων με απλή αρχιτεκτονική, που λόγω της αναπαράστασης bag-of-words δεν εκμεταλλεύονται σημαντικές πληροφορίες όπως η σειρά των λέξεων στο κείμενο.

Επίσης, το single-layer perceptron, παρά τον περιορισμό της γραμμικής διαχωρισιμότητας, πετυχαίνει καλύτερο Accuracy από το πιο πολύπλοκο δίκτυο με κρυφό επίπεδο. Αυτό ενδεχομένως συμβαίνει γιατί αν υποθέσουμε ότι οι λέξεις που συνεισφέρουν στην ταξινόμηση του κειμένου ως θετικού ή αρνητι-

Είδος μοντέλου	Θετικά σχόλια		Αρνητικά σχόλια		Συνολικά
	Precision	Recall	Precision	Recall	Accuracy
single-layer perceptron λέξεις + bigrams epochs: 20	93.6%	94.1%	90.5%	89.8%	**92.4%**
three-layer network λέξεις + bigrams epochs: 10 hidden layer size: 20	92.0%	94.2%	90.3%	86.9%	91.4%
three-layer network λέξεις + bigrams epochs: 20 hidden layer size: 50	92.7%	94.2%	90.5%	88.2%	91.9%
Naive Bayes λέξεις + bigrams (stems)	95.3%	95.8%	93.2%	92.5%	**94.5%**

Πίνακας 4.17: Αποτελέσματα παραλλαγών Νευρωνικών Δικτύων—Σύνολο κριτικών ξενοδοχείων

κού είναι μονοδιάστατα θετικές ή αρνητικές, ανεξάρτητα των συμφραζομένων, τότε η γραμμική διαχωρισιμότητα δεν μας περιορίζει. Μπορούμε να σκεφτόμαστε ότι οι θετικές λέξεις έχουν θετικά βάρη στο perceptron που δίνει την έξοδο για τη θετική γνώμη και αρνητικά βάρη στο perceptron που δίνει την έξοδο της αρνητικής γνώμης, και αντίστοιχα για τις θετικές λέξεις. Αυτού του τύπου το μοντέλο θα ήταν γραμμικά διαχωρίσιμο, αλλά αποτελεσματικό, παρότι επίσης καταδεικνύει ότι το single-layer perceptron λογικά δεν κερδίζει πολύ περισσότερο από το Naive Bayes.

Όσον αφορά το three-layer δίκτυο, και τη συνεισφορά του μεγέθους του κρυφού επιπέδου στο αποτέλεσμα: Δεν είδαμε βελτίωση με μεγαλύτερα κρυφά επίπεδα, κάτι που δείχνει ότι εάν το Νευρωνικό Δίκτυο εφαρμόζεται σε αναπαράσταση bag-of-words δεν υπάρχει κέρδος για τις βαθύτερες αρχιτεκτονικές.

Τέλος, σημειώνουμε ότι για την εκπαίδευση των νευρωνικών δικτύων χρειάζεται περισσότερος χρόνος εκτέλεσης σε σχέση με τον αλγόριθμο Naive Bayes. Ενώ ο τελευταίος ολοκληρώνει το 10-fold cross-validation σε μερικά δευτερόλεπτα, η τάξη μεγέθους του χρόνου εκτέλεσης για τα νευρωνικά δίκτυα κυμαίνεται από 4 λεπτά (single-layer perceptron) μέχρι πάνω από 30 λεπτά (three-layer network, με 50 νευρώνες στο κρυφό επίπεδο, εκπαίδευση για 20 εποχές). Φυσικά αυτά τα νούμερα εξαρτώνται και από την υλοποίησή μας, όσο και από το μέγεθος του συνόλου δεδομένων. Καταδεικνύουν όμως τη δυσκολία χρήσης που παρουσιάζουν τα νευρωνικά δίκτυα, αφού περιορίζεται το πλήθος των πειραμάτων που είναι εύκολο να γίνουν.

4.5.2 Δοκιμή σε κριτικές ταινιών

Ανάλογες δοκιμές με τη μέθοδο 10-fold cross-validation πραγματοποιήθηκαν και στο σύνολο κριτικών ταινιών των Pang και Lee. Επίσης παρατίθεται για σύγκριση και το καλύτερο μοντέλο Naive Bayes. Τα αποτελέσματα φαίνονται στον πίνακα 4.18.

Είδος μοντέλου	Θετικές κριτικές		Αρνητικές κριτικές		Συνολικά
	Precision	Recall	Precision	Recall	Accuracy
single-layer perceptron λέξεις + bigrams epochs: 20	81.2%	79.1%	79.6%	81.7%	**80.4%**
three-layer network λέξεις + bigrams epochs: 20 hidden layer size: 20	74.8%	89.0%	86.4%	70.0%	79.5%
Naive Bayes λέξεις + bigrams συχνότητα λέξεων >= 2	90.6%	74.5%	78.4%	92.3%	**83.4%**

Πίνακας 4.18: Αποτελέσματα παραλλαγών Νευρωνικών Δικτύων—Σύνολο κριτικών ταινιών

Ως προς την ορθότητα, παρατηρούμε και πάλι πως το απλούστερο δίκτυο (single-layer perceptron) έχει μεγαλύτερη τιμή (80.4%) από το δίκτυο τριών επιπέδων. Αλλά και πάλι, ακόμα μεγαλύτερη ορθότητα είχε μετρηθεί για το μοντέλο Naive Bayes. Έτσι, και σε αυτό το σύνολο δεδομένων ισχύουν τα ίδια συμπεράσματα με την προηγούμενη υποενότητα.

Επίσης παρατηρήσαμε ότι το Three-Layer Network δεν συνέκλινε πάντα σε λύση, αλλά κάποιες φορές εκφυλιζόταν σε μια κατάσταση όπου αποφαινόταν μονίμως την ίδια κατηγορία. Αναφέρουμε το αποτέλεσμα από παράδειγμα εκπαίδευσης όπου αυτό δε συνέβη.

Ως προς το χρόνο εκτέλεσης, επειδή το σύνολο αυτό αποτελείται από μεγαλύτερα σε έκταση κείμενα, είχαμε μεγαλύτερους χρόνους εκτέλεσης: Της τάξης των 10 λεπτών για το single-layer perceptron (για τις 10 δοκιμές του cross-validation), ενώ χρόνος σχεδόν 2 ωρών χρειάστηκε για το three-layer δίκτυο.

Κεφάλαιο 5

Συμπεράσματα

5.1 Σύνοψη αυτής της εργασίας

Με αφορμή τη συγκέντρωση ενός νέου συνόλου κειμένων ταξινομημένων ως προς θετική/αρνητική γνώμη, με πηγή την ιστοσελίδα κριτικών ξενοδοχείων booking.com, δοκιμάσαμε μια σειρά από μεθόδους αυτόματης ανάλυσης άποψης που βασίζονται στη μηχανική μάθηση. Στο πρόβλημα της εύρεσης της άποψης σε επίπεδο συνολικού κειμένου, υλοποιήσαμε και εφαρμόσαμε δύο μεθόδους, το μοντέλο Naive Bayes και Νευρωνικά Δίκτυα απλής αρχιτεκτονικής. Στα πειράματα που διεξήχθησαν διαπιστώθηκε πολύ υψηλή ορθότητα (94.5%) του μοντέλου Naive Bayes στο dataset κριτικών ξενοδοχείου, όταν ενσωματώθηκαν bigrams στην αναπαράσταση του κειμένου, και όταν οι λέξεις κανονικοποιούνταν στο θέμα τους (stem). Αυτό το ποσοστό όμως καθορίζεται σε μεγάλο βαθμό από εγγενή μορφολογικά στοιχεία του συνόλου δεδομένων. Το μοντέλο Naive Bayes επίσης δοκιμάστηκε στο standard dataset για τον τομέα της Ανάλυσης Άποψης, το σύνολο κριτικών ταινιών από την ιστοσελίδα imdb.com, των Pang και Lee. Σε αυτό το σύνολο καταγράφεται ορθότητα 83.4%. Στα πλαίσια της χρήσης του μοντέλου Naive Bayes, παρατηρήθηκαν οι παράμετροι που αυτό είχε μάθει ώστε να εξαχθούν συμπεράσματα για το πως αυτό δουλεύει, αλλά και να διαπιστωθούν χαρακτηριστικά των συνόλων δεδομένων.

Επίσης, κινηθήκαμε προς την κατεύθυνση και ενός λίγο διαφορετικού, και απαιτητικότερου προβλήματος, αυτό της ανάλυσης άποψης για συγκεκριμένες εκφάνσεις (Aspect-based Sentiment Analysis). Στόχος μας ήταν να εντοπίζουμε στο κείμενο συγκεκριμένες λέξεις, τόσο έκφρασης θετικής/αρνητικής γνώμης, όσο και αναφοράς σε διαφορετικές πλευρές του υπό κριτική αντικειμένου, στη συγκεκριμένη περίπτωση του ξενοδοχείου, εκφάνσεις όπως Εξυπηρέτηση, Προσωπικό, Κτιριακές Εγκαταστάσεις και Τοποθεσία. Το μοντέλο που χρησιμοποιήθηκε ήταν μια παραλλαγή του Hidden Markov Model που ονομάζεται Lexicalized Hidden Markov Model Integrating Part-of-Speech. Πρόκειται

για επαύξηση του τυπικού κρυφού μοντέλου Markov, που αξιοποιεί γλωσσικά στοιχεία όπως το μέρος του λόγου αλλά και λεκτικά μοτίβα, για να εντοπίσει τις ζητούμενες οντότητες μέσα στο κείμενο. Το μοντέλο αυτό εκπαιδεύτηκε και δοκιμάστηκε σε υποσύνολο των δεδομένων που είχαμε συλλέξει, το οποίο σημειώθηκε χειρονακτικά με ετικέτες σε επίπεδο λέξεων.

Διερευνήθηκαν οι παράμετροι του μοντέλου αυτού ως προς ποιες δίνουν την καλύτερη απόδοση. Στην καλύτερη περίπτωση, η απόδοση του μοντέλου στο να εντοπίζει της ετικέτες, παρουσίαζε precision 79.8% και recall 54.8%. Διαπιστώσαμε ότι τόσο το μέρους του λόγου των λέξεων όσο και η εξάρτηση από την προηγούμενη λέξη (lexicalized) παίζουν σημαντικό ρόλο στη διαμόρφωση της απόδοσης. Συγκρίνοντας τη συνεισφορά διαφορετικών επισημειωτών μέρους του λόγου (part-of-speech tagger), αποφανθήκαμε ότι ένας επισημειωτής που παράγει λιγότερες και απλούστερες κατηγορίες μέρους του λόγου δουλεύει καλύτερα.

Έγινε μια προσπάθεια για εφαρμογή Νευρωνικών Δικτύων στην ανάλυση άποψης σε επίπεδο συνολικού κειμένου. Δοκιμάστηκαν δύο απλές αρχιτεκτονικές, το perceptron ενός επιπέδου, και ένα δίκτυο τριών επιπέδων (δηλαδή με ένα κρυφό επίπεδο). Έχοντας ως είσοδο μια bag-of-words αναπαράσταση του κειμένου εισόδου, επαυξημένη με bigrams, και οι δύο αυτοί τύποι νευρωνικών δικτύων δεν κατάφεραν να ξεπεράσουν την απόδοση του Naive Bayes, τόσο στο σύνολο δεδομένων κριτικών ξενοδοχείων, όσο και στο σύνολο κριτικών ταινιών. Μάλιστα, η απλούστερη των δύο αρχιτεκτονικών είχε καλύτερη απόδοση.

Συμπεραίνουμε ότι για ανάλυση άποψης σε επίπεδο ολόκληρου κειμένου η μέθοδος Naive Bayes είναι αποτελεσματική, με επιπλέον πλεονεκτήματα ευκολία υλοποίησης, ταχύτητα και δυνατότητα επισκόπησης των παραμέτρων που έχει μάθει. Για να προτιμηθεί μια λύση με νευρωνικά δίκτυα, θα πρέπει να έχει ξεκάθαρο πλεονέκτημα απόδοσης, για να ισοφαριστούν τα μειονεκτήματα της αργής εκπαίδευσης, της μεγαλύτερης πολυπλοκότητας στην υλοποίηση και της αδυναμίας επισκόπησης των παραμέτρων. Όμως, η (ομολογουμένως απλοϊκή) εφαρμογή νευρωνικών δικτύων που δοκιμάσαμε δεν απέφερε καλύτερη απόδοση.

5.2 Ποιοτική σύγκριση αλγορίθμων

Ο πίνακας 5.1 συνοψίζει τις διαφορές μεταξύ των τριών μεθόδων που δοκιμάστηκαν. Με + σημειώνονται τα προτερήματα μιας μεθόδου σε μια κατηγορία, με - τα ελαττώματα, ενώ σημειώνεται ++ όταν ο αλγόριθμος αποδίδει εξαιρετικά καλά.

Χαρακτηριστικό	Naive Bayes	HMM	Νευρωνικά Δίκτυα
Precision	++	+	+
Recall	++	-	+
Χρόνος Εκπαίδευσης	++	++	-
Χρόνος Εφαρμογής	++	+	++
Aspect-Based	-	+	-
Ευκολία Υλοποίησης	++	-	-
Απαιτεί Manual Tagging	+ (ΟΧΙ)	- (ΝΑΙ)	+ (ΟΧΙ)
Μεταφερσιμότητα Domain	+	-	+

Πίνακας 5.1:

Βιβλιογραφία

[Bas+12] Frédéric Bastien et al. *Theano: new features and speed improvements*. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop. 2012.

[Ber+10] James Bergstra et al. "Theano: a CPU and GPU Math Expression Compiler". In: *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Oral Presentation. Austin, TX, June 2010.

[BKL09] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

[GB10] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *International conference on artificial intelligence and statistics*. 2010, pp. 249–256.

[JHS09] Wei Jin, Hung Hay Ho, and Rohini K Srihari. "OpinionMiner: a novel machine learning system for web opinion mining and extraction". In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, pp. 1195–1204.

[Koh+95] Ron Kohavi et al. "A study of cross-validation and bootstrap for accuracy estimation and model selection". In: *Ijcai*. Vol. 14. 2. 1995, pp. 1137–1145.

[MP43] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[Nta06] Georgios Ntais. "Development of a Stemmer for the Greek Language". PhD thesis. Royal Institute of Technology, 2006.

[Pic94] Phil Picton. *Introduction to neural networks*. Macmillan Publishers Limited, 1994.

[PL04] Bo Pang and Lillian Lee. "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In: *Proceedings of the ACL*. http://www.cs.cornell.edu/people/pabo/movie-review-data/. 2004.

[PL08] Bo Pang and Lillian Lee. "Opinion mining and sentiment analysis". In: *Foundations and trends in information retrieval* 2.1-2 (2008). http://www.cs.cornell.edu/home/llee/opinion-mining-sentiment-analysis-survey.html, pp. 1–135.

[PLV02] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques". In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics. 2002, pp. 79–86.

[Por80] Martin F Porter. "An algorithm for suffix stripping". In: *Program* 14.3 (1980), pp. 130–137.

[RN] Stuart Russell and Peter Norvig. *Artificial intelligence: a modern approach (2nd edition)*. Prentice Hall.

[Ros58] Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[Sar09] Spyridon Saroukos. "Enhancing a greek language stemmer-efficiency and accuracy improvements". In: (2009).

[SB07] Benjamin Snyder and Regina Barzilay. "Multiple Aspect Ranking Using the Good Grief Algorithm." In: 2007.

[Tur02] Peter D Turney. "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics. 2002, pp. 417–424.

[W+60] Bernard Widrow, Marcian E Hoff, et al. "Adaptive switching circuits". In: *IRE WESCON Convention Record*. 1960, pp. 96–104.

[WCV11] S. van der Walt, S.C. Colbert, and G. Varoquaux. "The NumPy Array: A Structure for Efficient Numerical Computation". In: *Computing in Science Engineering* 13.2 (Mar. 2011), pp. 22–30. ISSN: 1521-9615. DOI: 10.1109/MCSE.2011.37.

[Βλα+11] Ιωάννης Βλαχάβας et al. *Τεχνητή Νοημοσύνη - Γ' Έκδοση*. Εκδόσεις Πανεπιστημίου Μακεδονίας, 2011.