Πολυτεχνειο Κρητης

Σχολη Ηλεκτρονικων Μηχανικων και Μηχανικων Υπολογιστων

Ομαδοποίηση και Βελτιστοποίηση σε Συνεχή Γλωσσικά Μοντέλα

Κωνσταντίνος Πεχλιβάνης

Εξεταστική Επιτροπή

Καθ. Βασίλειος Διγαλάκης (ΗΜΜΥ)

Αναπλ. Καθ. Μιχαήλ Γ. Λαγουδάκης (ΗΜΜΥ)

Καθ. Μιχαήλ Πατεράκης (ΗΜΜΥ)

Χανιά, Δεκέμβριος 2013

Technical University of Crete

School of Electronic and Computer Engineering

Clustering and Optimization in Continuous Language Models

Konstantinos Pechlivanis

Thesis Committee

Professor Vasilios Digalakis (ECE)

Associate Professor Michael Lagoudakis (ECE)

Professor Michael Paterakis (ECE)

Chania, December 2013

"A C program is like a fast dance on a newly waxed dance floor by people carrying razors." (Waldi Ravens)

Περίληψη

Στην εργασία αυτή, ασχολούμαστε με την κατασκευή στατιστικών γλωσσικών μοντέλων στο συνεχή χώρο χρησιμοποιώντας ομαδοποιημένα μείγματα κανονικών κατανομών. Τα κίνητρα για τη προσέγγιση αυτή είναι ότι σε πρώτο στάδιο μπορούν να υπολογιστούν μη μηδενικές πιθανότητες, για ακολουθίες λέξεων, ακόμα και αν δεν έχουν βρεθεί στα δείγματα εκπαίδευσης (εξομάλυνση). Σε δεύτερο στάδιο, μας παρέχεται ένα πολύ πιο ευέλικτο πλαίσιο αλγορίθμων το οποίο μπορεί να μας βοηθήσει στη βελτίωση και προσαρμογή του μοντέλου.

Στο κύριο μέρος της εργασίας επικεντρωνόμαστε στην εφαρμογή διαφορετικών τεχνικών ομαδοποίησης των συνεχών κατανομών, μέσω της ομαδοποίησης των λέξεων του λεξιλογίου με αρχική αντιστοίχηση των λέξεων σε διανύσματα. Οι τεχνικές χωρίζονται σε τρεις κύριες οικογένειες. Αυτές που βασίζονται στην ελαχιστοποίηση του perplexity (χρήση αλγορίθμου Brown), τις ιεραρχικές (συσσωρευτικοί και διαιρετικοί αλγόριθμοι), και αυτές που βασίζονται στη βελτιστοποίηση της συνάρτησης κόστους (αυστηρός αλγόριθμος ομαδοποίησης, K-μέσων). Οι ομαδοποιήσεις που λαμβάνουν χώρα στηρίζονται στην λεξιλογική και σημασιολογική ομοιότητα των λέξεων. Το επόμενο βήμα, αφορά την εκτίμηση του πίνακα συν-εμφανίσεων των κλάσεων, ο οποίος στη συνέχεια συμπιέζεται με στόχο τη μείωση των διαστάσεων και κανονικοποίησης των τιμών του, κάνοντας χρήση της μεθόδου, Singular Value Decomposition (SVD). Κατά το επόμενο στάδιο, συγκεντρώσαμε τα ιστορικά διανύσματα κάθε κλάσης τα οποία χρησιμοποιήσαμε σε πρώτη φάση για την εκτίμηση στατιστικών τιμών με σκοπό την προβολή αυτών των διανυσμάτων σε χαμηλότερη διάσταση με τη χρήση της τεχνικής μείωσης διαστάσεων, Linear Discriminant Analysis (LDA). Σε δεύτερο φάση, εκμεταλλευόμαστε αυτά τα ιστορικά διανύσματα που έχουμε προβάλει, ώστε να εκπαιδεύσουμε κάθε κλάση πάνω σε συνεχή μίξη Κανονικών κατανομών τις οποίες επεκτείναμε χρησιμοποιώντας μίξη «δεμένων» Κανονικών κατανομών. Το τελευταίο κομμάτι υλοποίησης, αφορά τη χρήση κάποιων μονάδων μέτρησης (εντροπία, περιπλοκή), από τη Θεωρία Πληροφορίας, για την αξιολόγηση της ακρίβειας των μοντέλων που κατασκευάσαμε. Πριν την εκτέλεση αριθμού πειραμάτων, με σκοπό την εύρεση των παραμέτρων που οδηγούν στο όσο το δυνατόν πιο εύρωστο/ακριβές γλωσσικό μοντέλο εφαρμόσαμε μια σειρά από βελτιστοποιήσεις.

Η βελτιστοποίηση περιλαμβάνει τη χρήση πιο γρήγορων γλωσσών προγραμματισμού (C) για την υλοποίηση κομματιών κώδικα που έχουν μεγάλη υπολογιστική πολυπλοκότητα και κατ΄ επέκταση μεγάλο χρόνο εκτέλεσης. Επιπλέον, κάνουμε εκτενή χρήση των hash tables και εφαρμόζουμε βελτιστοποιήσεις στον αλγόριθμο. Το αποτέλεσμα ήταν η σημαντική βελτίωση του χρόνου εκτέλεσης του αλγορίθμου.

Τα πειράματά μας για την εκτίμηση του μοντέλου έγιναν σε δεδομένα της εφημερίδας

WSJ, τα αποτελέσματα των οποίων δείχνουν ότι χρησιμοποιώντας τεχνικές ομαδοποίησης μπορούμε να μειώσουμε σημαντικά την πολυπλοκότητα (ppl) του συνεχούς γλωσσικού μοντέλου.

Λέξεις κλειδιά: ομαδοποίηση, βελτιστοποίηση, συνεχή γλωσσικά μοντέλα, πίνακας συν-εμφανίσεων, λεξιλογική-σημασιολογική-συντακτική ομοιότητα, ιεραρχικοί αλγόριθμοι ομαδοποίησης, SRILM, k-means, GAAG, rbr, direct, graph/KNN, bagglo, SVD, χαρακτηριστικά διανύσματα, διανύσματα λέξεων, αντιστοίχηση κλάσεων/λέξεων σε διανύσματα, διανύσματα ιστορικών, n-grams, LDA, προβολή, εκπαίδευση, μίξη Gaussian μοντέλων, μίξη δεμένων Gaussian μοντέλων, εντροπία, περιπλοκή.

Abstract

In this thesis, we are concerned with the construction of statistic language models in continuous space, using clustered mixtures of uniform distributions. The initial motivations for this approach are the capability of computing non-zero probabilities for tokens, even if they are unseen events in the training samples (smoothing). Secondly, we are provided by a more flexible variety of algorithms, that can help us in the improvement and adaptation of the model.

In the main concept of thesis, we are focused on clustering the words of the vocabulary using a variety of different clustering techniques which are based in word mapping. The techniques are distinguished in three main categories. Firstly, the ones that are based on the minimization of the perplexity (using Brown algorithm), the hierarchical algorithms (agglomerative and divisive algorithms), and the ones that are based on the optimization of cost function (hard/crisp clustering algorithm, K-means). Clustering, which is taking place, is based on the lexical and semantic similarity of data. Thereafter, we estimated the class co-occurrence matrix, which is compressed in order to reduce the dimensions and achieve normalization on the values, using the SVD technique. Then, we used the history vectors of each class to assess statistical values in order to project these vectors in a lower dimension, using the LDA technique. The second phase, regards the exploitation of these projected history vectors in order to train each class into Gaussian Mixture Models, which was extended using Tied Gaussian Mixtures Models. Finally, we used measurements like entropy and perplexity (Information Theory), to assess the accuracy of the models manufactured. Before executing the experiments, which lead on finding the parameters of a more robust language model, we applied a number of optimizations.

The optimization includes the use of faster programming languages (C), in specific parts of our implementation which have high computational complexity and as a result a large execution time. In addition, we made an extensive use of hash tables and optimizations of the algorithm. The consequence was a significant improvement of the algorithm's execution time.

Our experiments for the estimation of the model, were applied on data of WSJ newspaper, whose results show that the use of clustering techniques can significantly reduce the complexity (ppl) of the continuous language model.

keywords: clustering, optimization, continuous language model, co-occurrence matrix, lexical-semantic-syntactic similarity, hierarchical clustering algorithms, SRILM, k-means, GAAG, rbr, direct, graph/KNN, bagglo, SVD, feature vectors, word vectors, word mapping, history vectors, n-grams, LDA, projection, training, Gaussian mixture models,

tied Gaussian mixture models, entopy, perplexity.

Ευχαριστίες

Πρώτα απ΄ όλα, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου, κύριο Βασίλη Διγαλάκη για την συνεργασία και τη υποστήριξη προς το πρόσωπο μου, δίνοντας μου τη δυνατότητα να ασχοληθώ με αυτόν τον εξαιρετικά ενδιαφέρον τομέα της Επιστήμης των Υπολογιστών που ελπίζω να ακολουθήσω και στο μέλλον, καθώς και το κύριο Βασίλη Διακολουκά για τις αμέτρητες συμβουλές και βοήθειες καθ΄ όλη τη διάρκεια υλοποίησης της διπλωματικής.

Επίσης, θα ήθελα να ευχαριστήσω μέσα από τη καρδιά μου, τους φίλους μου, για όλες αυτές τις μοναδικές και ανεκτίμητες στιγμές που περάσαμε όλα αυτά τα χρόνια. Η συναισθηματική και ψυχολογική υποστήριξή τους ήταν η μεγαλύτερη πηγή δύναμης. Θα συντροφεύεται πάντα τις αναμνήσεις μου, Κώστας Καράλας, Γιάννης Λιβέριος, Βαγγέλης Μιχελουδάκης, Αλέξανδρος Μαυρομμάτης, Στέλλα Μαροπάκη, Σωτήρης Σουρλαντζής, Γιάννης Ταμπακάκης, Αροδάμη Χωριανοπούλου.

Τέλος, πάνω από όλα θα ήθελα να ευχαριστήσω τους γονείς μου, για τις θυσίες, την υπομονή και την στήριξη τους όλα αυτά τα χρόνια, χωρίς αυτούς τίποτα απ΄ όλα αυτά δεν θα είχε επιτευχθεί.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κατάλογος Αλγορίθμων

Κεφάλαιο 1

Εισαγωγή

Η γλώσσα είναι το κύριο μέσο του ανθρώπου για τη διατύπωση σκέψεων και την ανταλλαγή απόψεων, γνώσεων και πληροφοριών, καθώς και τη μετάδοσή τους από γενιά σε γενιά μέσα στους αιώνες. Στις μέρες μας, λόγω του διαδικτύου, ο όγκος των γραπτών δεδομένων που μπορεί να έχει πρόσβαση κανείς είναι πλέον τεράστιος. Συνεπώς, είναι επιτακτική η ανάγκη να υπάρξει ένας τρόπος έτσι ώστε να μπορέσουμε να επωφεληθούμε από όλα αυτά τα δεδομένα, εξάγοντας τα στοιχεία εκείνα που μας ενδιαφέρουν και μας αφορούν, χωρίς όμως να χρειαστεί να ξοδέψουμε πολύ από το χρόνο που διαθέτουμε διαβάζοντάς τα ενδελεχώς.

Η ομαδοποίηση (clustering) αποτελεί αναπόσπαστο κομμάτι στην διερεύνηση της εξόρυξης δεδομένων (data mining), όντας πολύ διαδεδομένη τεχνική για τη στατιστική ανάλυση δεδομένων που χρησιμοποιείται σε διάφορους τομείς, όπως μηχανική μάθηση (machine learning), αναγνώριση προτύπων (pattern recognition), ανάλυση εικόνας (image analysis), ανάκτηση πληροφοριών (information retrieval) και βιοπληροφορική (bioinformatics). Στη παρούσα διπλωματική, θα επικεντρωθούμε στην επεξεργασία φυσική γλώσσας (NLP) εκμεταλλευόμενοι την ομαδοποίηση λέξεων (word clustering). Πιο συγκεκριμένα, word clustering είναι η κατηγοριοποίηση των λέξεων ενός η περισσότερων κειμένων σε ομάδες με τρόπο τέτοιο, ώστε η κάθε ομάδα-κλάση να συγκεντρώνει τις λέξεις εκείνες, που έχουν τα περισσότερο όμοια χαρακτηριστικά (features). Στη γλωσσολογία, είναι σύνηθες φαινόμενο η ταξινόμηση αυτών των χαρακτηριστικών σε τρεις ευρείες κλάσεις κατηγοριών: συνταχτική (syntactic), σημασιολογική (semantic), και λεξικολογική (lexical). Με χρήση κατά κύριο λόγο lexical χαρακτηριστικών, επιχειρούμε μεγάλο αριθμό διαφορετικών τεχνικών word clustering στοχεύοντας στην βελτίωση της ακρίβειας των γλωσσικών μοντέλων (language model) που υλοποιήθηκαν πάνω σε μείγματα από Gaussian κατανομές και μείγματα δεμένων

Gaussian μοντέλων (Tied Gaussian Mixture Models) στο συνεχή χώρο.

1.1 Σκοπός Διπλωματικής

Μέσα από αυτή τη διπλωματική, προτείνουμε βελτιωμένα γλωσσικά μοντέλα με χρήση ομαδοποίησης, που χρησιμοποιούν συνεχείς κατανομές, και πιο συγκεκριμένα Gaussian κατανομές, Gaussian Mixture Models και τις υλοποιήσεις τους, που είναι οι Tied Gaussian Mixture Models, όπου συγχρόνως το αρχικό μοντέλο λαμβάνει υπόψη, όχι κάθε λέξη ξεχωριστά, αλλά κλάσεις λέξεων. Αρχικά, η ιδέα για συνεχή γλωσσικά μοντέλα δεν είναι τελείως καινούρια, αφού οι Afify et al. [1] και ο Chen [2], πρότειναν πρώτοι γλωσσικά μοντέλα με Gaussian mixtures για αναγνώριση φωνής. Επίσης, εκτενείς προσπάθειες πάνω σε αυτού του είδους τα μοντέλα πραγματοποιήθηκαν σε παλιότερη διπλωματική [3], σε γλωσσικά μοντέλα όπου η είσοδος των δεδομένων γινόταν λαμβάνοντας υπόψη κάθε λέξη ξεχωριστά και χρησιμοποιώντας ένα περιορισμένο σε μέγεθος λεξιλόγιο. Η χρήση γλωσσικών μοντέλων στο συνεχή χώρο αποσκοπεί στα πλεονεκτήματα που παρουσιάζουν τα μοντέλα αυτά, όπως είναι η εξομάλυνση των γεγονότων που δεν έχουν εμφανιστεί-παρατηρηθεί (smoothing unseen events) και η ευκολία προσαρμογής (ease of adaptation).

Ωστόσο, απ΄ όσο γνωρίζουμε, ανάλογες τεχνικές συνεχούς χώρου δεν έχουν εφαρμοστεί μέχρι στιγμής, σε συνδυασμό με εκτέλεση διαφορετικών τεχνικών clustering πάνω στα δεδομένα εισόδου. Επιπλέον, αυτές οι παλιότερες υλοποιήσεις αφορούν τη χρήση περιορισμένου σε μέγεθος λεξιλογίου (που μοντελοποιεί τις λιγότερο συχνές λέξεις μέσω της κλάσης unk), και όχι του συνόλου των διαφορετικών λέξεων του κειμένου, κάτι που επίσης αποτελεί μια πρώτη προσπάθεια υλοποίησης.

Συνεχίζοντας, χρειαζόμαστε ένα μοντέλο το οποίο να μην είναι ούτε πολύ απλό, για να μπορεί να προσεγγίσει την πραγματική κατανομή, ούτε όμως και τόσο περίπλοκο, για να μπορεί να εκπαιδευτεί και να ελεγχθεί όσο το δυνατόν πιο αποδοτικά. Οπότε, κατά τη διαδικασία της εκπαίδευσης, χρειάστηκε αρκετός πειραματισμός εξετάζοντας διαφορετικές παραμέτρους, έτσι ώστε να καταλήξουμε τελικά σε μια στρατηγική που μας αποφέρει καλά αποτελέσματα. Δομικό στοιχείο στην όλη διαδικασία, αποτελεί η επιλογή των κατάλληλων τεχνικών clustering και το μέγεθος του λεξιλογίου που έχει επιλεχθεί. Προέκυψαν ενδιαφέροντα αποτελέσματα τα οποία θα αναλύσουμε στη συνέχεια της εργασίας.

1.2 Συνεισφορά Διπλωματικής

Αυτή η διπλωματική εργασία αποτυπώνει την έρευνα που έγινε για τη βελτιστοποίηση της διαδικασίας εκπαίδευσης συνεχών γλωσσικών μοντέλων, όπου το εκάστοτε λεξιλόγιο έχει ομαδοποιηθεί χρησιμοποιώντας διαφορετικές τεχνικές. Πιο συγκεκριμένα, προσπαθούμε να διαπιστώσουμε τις επιπτώσεις που έχει η εφαρμογή clustering στα δεδομένα εισόδου για τον εν λόγω τομέα, καθώς επίσης και το αν μπορούμε να επωφεληθούμε από τις ιδιότητες που διέπουν το χώρο αυτό, για να πετύχουμε κάποια καλύτερα αποτελέσματα σε σχέση με αυτά που έχουν παρουσιαστεί κατά το παρελθόν και προέρχονται από τον χειρισμό των δεδομένων εισόδου λαμβάνοντας υπόψη κάθε λέξη ξεχωριστά και με εφαρμογή περιορισμένου λεξιλογίου.

Διερευνήσαμε διαφορετικούς αλγορίθμους ομαδοποίησης των λέξεων καθώς επίσης και διαφορετικές μεθόδους εφαρμογής τους στα πλαίσια της διαδικασίας εκπαίδευσης των συνεχών γλωσσικών μοντέλων. Χρησιμοποιήσαμε αλγορίθμους ομαδοποίησης που αποσκοπούν στην ελαχιστοποίηση του perplexity κάνοντας χρήση του αλγορίθμου Brown. Επίσης, εφαρμόσαμε ιεραρχικούς αλγορίθμους δίνοντας έμφαση σε συσσωρευτικούς και διαιρετικούς αλγορίθμους καθώς και αυτούς που βασίζονται στη βελτιστοποίηση της συνάρτησης κόστους, εμαρμόζοντας τον αυστηρό αλγόριθμος ομαδοποίησης, K-means. Οι ομαδοποιήσεις που λαμβάνουν χώρα στηρίζονται στην λεξιλογική και σημασιολογική ομοιότητα των λέξεων και εφαρμόζονται πάνω στις συνηθέστερες λέξεις στη περίπτωση χρήσης του μικρού λεξιλογίου και σε όλο το λεξιλόγιο στη περίπτωση χρήσης όλων των διαφορετικών λέξεων.

Η επιλογή ομαδοποίησης των λέξεων, αποσκοπεί στη καλύτερη μοντελοποίηση των δεδομένων εισόδου. Πιο συγκεκριμένα, ομαδοποιώντας όλες τις λέξεις επιτυγχάνουμε πολύ καλύτερη μοντελοποίηση των λέξεων που δεν έχουμε αρκετή πληροφορία, δηλαδή των λέξεων που εμφανίζονται ελάχιστες φορές στα δεδομένα μας. Έτσι, ομαδοποιώντας «σπάνιες» λέξεις με άλλες συχνότερες λέξεις επιτυγχάνουμε πολύ καλύτερη εκπαίδευση των λέξεων αυτών που σε διαφορετική περίπτωση η εκπαίδευση τους θα ήταν αναποτελεσματική.

Ακόμη, γίνεται φανερό ότι η χρήση τεχνικών και μεθόδων μείωσης διαστάσεων, Singular Vectors Decomposition-SVD που συρρικνώνει και τροποποιεί τα διανύσματα των λέξεων, και Linear Descriminant Decomposition - LDA που προβάλει (projection) τα δεδομένα στη μειωμένη διάσταση, επιφέρουν σημαντικές βελτιώσεις τόσο στα αποτελέσματα όσο και στη πολυπλοκότητα της υλοποίησης.

Η τεχνική clustering που απέφερε γενικά καλύτερα αποτελέσματα ήταν μέσω του εργαλείου SRILM (χρήση του αλγορίθμου Brown για την ομαδοποίηση των δεδομένων) η οποία

βελτιώνει σημαντικά τα αποτελέσματα παλιότερων υλοποιήσεων για τη χρήση μικρού μεγέθους λεξιλογίου. Η υλοποίηση που αφορά τη χρήση του σύνολου των λέξεων του κειμένου, ως λεξιλόγιο, τα αποτελέσματα σαν πρώτη σκέψη δεν είναι το ίδιο καλά, όπως επίσης δεν είναι συγκρίσιμα γιατί δεν έχει υλοποιηθεί αντίστοιχες δουλειές με τη χρήση τόσο μεγάλου λεξιλογίου σε συνεχή γλωσσικά μοντέλα.

Τέλος, αποτελέσματα που προέκυψαν και μέσω ορισμένων πειραμάτων που δοκιμάστηκαν πάνω στα γνήσια-ακατέργαστα δεδομένα χωρίς μείωση των διαστάσεων (SVD), τα αποτελέσματα δεν είναι το ίδιο καλά, καθώς επίσης και η χρονική και χωρική πολυπλοκότητα η οποία αυξάνεται σε αρκετά μεγάλα επίπεδα σε σχέση με την προηγούμενη προσέγγιση.

1.3 Περίγραμμα Διπλωματικής

Αρχικά, στο Κεφάλαιο 2 περιγράφουμε τα Στατιστικά Γλωσσικά Μοντέλα στο διακριτό χώρο δίνοντας έμφαση στα N-grams μοντέλα, ενώ εστιάζουμε στις δυσκολίες που παρουσιάζουν, αναλύοντας τα μειονεκτήματα των N-grams μοντέλων. Έπειτα, εισχωρούμε στη Θεωρία Πληροφορίας με σκοπό την αξιολόγηση της Ποιότητας των Γλωσσικών Μοντέλων, αναλύοντας τα μαθηματικά εργαλεία της Εντροπίας και του Perplexity που θα χρησιμοποιήσουμε για την αξιολόγηση αυτή. Στη συνέχεια, αναφέρουμε κάποιους γενικούς ορισμούς και βασικές έννοιες σχετικά με τη σημασία της ομαδοποίησης. Τέλος, κάνουμε μια ανασκόπηση της κοντινότερης δουλείας που έχει γίνει, συνοδευόμενη πάντα από τις αντίστοιχες αποδόσεις που έχουν επιτευχθεί σε σχέση με τη δική μας.

Το Κεφάλαιο 3 περιλαμβάνει το μεγαλύτερο κομμάτι υλοποίησης που πραγματοποιήθηκε. Αναλυτικότερα, περιγράφεται η διαδικασία εκτίμησης του πίνακα συν-εμφανίσεων κλάσεων (class co-occurence) και η μετέπειτα τεχνική μείωσης διαστάσεων, SVD, που εφαρμόζεται στο πίνακα αυτό. Αφού γίνει περιγραφή της εύρεσης των ιστορικών διανυσμάτων, αναλύουμε την τεχνική LDA, για την προβολή των διανυσμάτων σε χαμηλότερη διάσταση. Τέλος, υπάρχει η περιγραφή για τα μίγματα Gaussian μοντέλων πάνω στα οποία θα εκπαιδεύσουμε κάθε κλάση.

Στο Κεφάλαιο 4 που ακολουθεί, περιγράφουμε την διαδικασία της ομαδοποίηση (clustering) και ακόμα πιο συγκεκριμένα την ομαδοποίηση λέξεων (word clustering) για τις κλάσεις των διαφορετικών διαθέσιμων χαρακτηριστικών (syntactic, semantic, lexical). Επιπλέον, αναλύουμε κάθε μια ξεχωριστά τις τεχνικές clustering και τους αλγορίθμους που χρησιμοποιήθηκαν, καθώς και τις προπαρασκευαστικές διαδικασίες που απαιτούνται για την αντιστοίχιση κάθε λέξης του λεξιλογίου με κάποιο διάνυσμα (word mapping) με σκοπό την

μετέπειτα ομαδοποίηση τους.

Στο Κεφάλαιο 5 βρίσκονται τα αποτελέσματα που επιτύχαμε για τα δεδομένα μας. Στην αρχή του ίδιου κεφαλαίου, γίνεται λόγος τόσο για το dataset που χρησιμοποιήσαμε, το οποίο είναι τα coprus Wall Street Journal, όσο και για το σημείο εκκίνησής μας (baseline). Επίσης, γίνεται αναφορά στις βελτιστοποιήσεις που λαμβάνουν χώρα και στις διάφορες εναλλακτικές μεθόδους μοντελοποίησης των δεδομένων

Τέλος, στο Κεφάλαιο 6 σκιαγραφούμε τα συμπεράσματά τα οποία προκύπτουν και έχουμε καταλήξει, συγκρίνουμε τα αποτελέσματα με αυτά προηγούμενων μελετών και προτείνουμε μελλοντική περαιτέρω έρευνα που μπορεί να γίνει.

Κεφάλαιο 2

Θεωρητικό και Ιστορικό Υπόβαθρο

2.1 Γλωσσικά Μοντέλα

Το στατιστικό γλωσσικό μοντέλο (Stochastic Language Model-SLM), χρησιμοποιεί τεχνικές στατιστικής εκτίμησης γλωσσικών δεδομένων εκπαίδευσης, που εφαρμόζονται σε εκτεταμένα κείμενα, με σκοπό την μοντελοποίηση της γλώσσας. Ανάμεσα στις πιο δημοφιλείς τεχνικές στατιστικής εκτίμησης είναι και τα μοντέλα N-grams που αναλύουμε παρακάτω. Ο ρόλος τους είναι πολύ σημαντικός για μια σειρά από εφαρμογές της γλωσσικής τεχνολογίας, όπως η αναγνώριση φωνής, η οπτική αναγνώριση χαρακτήρων, η μηχανική μετάφραση ακόμη και η ορθογραφική διόρθωση

Οι προσπάθειες επικεντρώνονται στο να υπολογίσουμε με ακρίβεια την πιθανότητα P($\mathbf{W}$) για μια δεδομένη ακολουθία λέξεων $\mathbf{W} = w_1, w_2, ..., w_n$. Ο βασικός στόχος των SLM είναι να παράγει επαρκή πιθανοτικές πληροφορίες, έτσι ώστε οι πιο πιθανές ακολουθίες λέξεων να έχουν υψηλότερη πιθανότητα. Το πιο διαδεδομένο SLM είναι το N-gram μοντέλο. Ένα γλωσσικό μοντέλο μπορεί να διατυπωθεί ως μια κατανομή πιθανοτήτων P($\mathbf{W}$) πάνω σε strings λέξεων $\mathbf{W}$ που ανταναχλά το πόσο συχνά μια σειρά $\mathbf{W}$ εμφανίζεται ως πρόταση.

Χρησιμοποιώντας τον κανόνα της αλυσίδας η πιθανότητα P($\mathbf{W}$) μπορεί να αναλυθεί ως εξής:

$$\begin{aligned} P(\mathbf{W}) &= P(w_1, w_2, ..., w_n) \\ &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)...P(w_n|w_1, w_2, ..., w_{n-1}) \\ &= \prod_{i=1}^{n} P(w_i|w_1, w_2, ..., w_{i-1}) \end{aligned}$$

όπου $P(w_i|w_1, w_2, ..., w_{i-1})$ είναι η πιθανότητα να ακολουθεί η λέξη w_i, δεδομένης της ακολουθίας λέξεων που παρουσιάστηκε προηγουμένως. Η επιλογή του w_i εξαρτάται συνεπώς από ολόκληρο το ιστορικό της εισόδου. Για ένα λεξιλόγιο μεγέθους V υπάρχουν V^{i-1} διαφορετικά ιστορικά και έτσι, για να καθορίσουμε την πιθανότητα $P(w_i|w_1, w_2, ..., w_{i-1})$, τελείως, θα πρέπει να εκτιμηθούν V^i τιμές. Στην πραγματικότητα, οι πιθανότητες $P(w_i|w_1, w_2, ..., w_{i-1})$ είναι αδύνατον να εκτιμηθούν, ακόμη και για όχι για πολύ μεγάλες τιμές του i, δεδομένου ότι τα περισσότερα ιστορικά είναι μοναδικά ή έχουν συμβεί λίγες φορές. Μια πρακτική λύση στα παραπάνω προβλήματα είναι η χρήση Μαρκοβιανής υπόθεσης (Markovian assumption), θεωρώντας ότι ο όρος $P(w_i|w_1, w_2, ..., w_{i-1})$ εξαρτάται μόνο από ορισμένες κλάσεις ισοδυναμίας. Η κλάση ισοδυναμίας μπορεί απλά να βασίζεται στις διάφορες προηγούμενες λέξεις. Αυτό οδηγεί σε ένα N-gram μοντέλο γλώσσας που χρησιμοποιεί τις προηγούμενες N-1 λέξεις σε μια ακολουθία για να προβλέψει την επόμενη λέξη. Το N-gram μοντέλο υλοποιείται ως εξής :

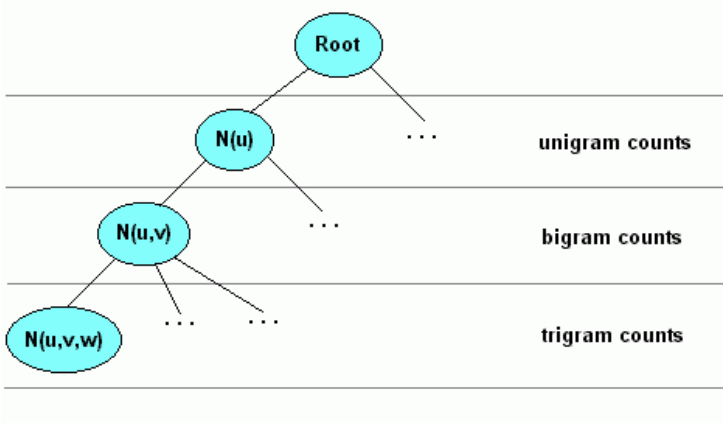
- Αν η πιθανότητα εμφάνισης της τρέχουσας λέξεις w_i, δεν εξαρτάται από καμιά προηγούμενη λέξη της ακολουθίας, τότε έχουμε ένα unigram: $P(w_i)$

- Αν η πιθανότητα εμφάνισης της τρέχουσας λέξεις w_i, εξαρτάται από την ακριβώς προηγούμενη λέξη της ακολουθίας, τότε έχουμε ένα bigram: $P(w_i|w_{i-1})$

- Αν η πιθανότητα εμφάνισης της τρέχουσας λέξεις w_i, εξαρτάται από τις ακριβώς δυο προηγούμενες λέξεις της ακολουθίας, τότε έχουμε ένα trigram: $P(w_i|w_{i-1}, w_{i-2})$

- Με την ίδια λογική ορίζουμε και τα 4-gram, 5-gram,...,N-gram

Τα trigram είναι ιδιαίτερα ισχυρά. Αυτό οφείλεται στο γεγονός ότι οι περισσότερες λέξεις έχουν ισχυρή εξάρτηση από τις δυο προηγούμενες, κάτι το οποίο μπορεί να εκτιμηθεί αρκετά καλά με κάποιο εφικτό corpus.



Σχήμα 2.1: N-grams

2.1.1 Περιγραφή και Μειονεκτήματα των N-grams

Ακόμα και τα πιο απλά μοντέλα μπορούν να επηρεάσουν την απόδοση στις εφαρμογές στις οποίες λαμβάνουν χώρα . Για παράδειγμα, τα γλωσσικά μοντέλα είναι πολύ ευαίσθητα στο να εκπαιδεύουν δεδομένα (train data) για εναλλαγή θέματος (topic alternation). Άλλο ενδιαφέρον παράδειγμα, είναι η δημιουργία ενός μοντέλου για καθημερινές τηλεφωνικές συνομιλίες το οποίο θα είναι πιο ακριβές εάν δυο εκατομμύρια από τέτοιες ηχογραφήσεις έχουν χρησιμοποιηθεί από 140 εκατομμύρια ηχογραφήσεις τηλεόρασης και ραδιοφώνου. Αυτή η επίδραση είναι τόσο ισχυρή, ακόμη και για αλλαγές που είναι απαραίτητες για την ανθρώπινη προοπτική. Ένα καλά δομημένο γλωσσικό μοντέλο που έχει εκπαιδευτεί με τα Wall Street Journal corpus (αποτελούν τα δεδομένα που θα χρησιμοποιήσουμε σε όλη τη διαδικασία της διπλωματικής) δεν θα μπορούσε να είναι ποτέ ακριβές/αποδοτικό εάν εφαρμόζεται σε ένα επιστημονικό ή αθλητικό corpus.

Όταν τα δεδομένα εκπαίδευσης (train data) διαφέρουν από τα δεδομένα δοκιμής (test data), η ποιότητα του εκπαιδευμένου μοντέλου δεν θα είναι τόσο ικανοποιητική. Ακόμα και σε μεγάλα σύνολα δεδομένων, θα υπάρχουν πολλά N-grams που δεν εμφανίζονται ποτέ. Όπως επίσης, είναι αναμενόμενο ότι πολλά «αποδεκτά» N-grams δεν θα εμφανίζονται στο σύνολο των train data. Επιπλέον, εάν οι σχετικές συχνότητες χρησιμοποιούνται για την εκτίμηση πιθανοτήτων, υπάρχει μεγάλη πιθανότητα να ληφθούν ανεπαρκείς εκτιμήσεις για τις μικρές σε αριθμό εμφανίσεις των N-grams. Τις περισσότερες φορές, είναι αναγκαίο να γίνει εξομάλυνση (smoothing) των N-grams πιθανοτήτων, ώστε να εμφανίζουν κανονικοποιημένες κατανομές. Πολλές τεχνικές smoothing έχουν αναπτυχθεί κατά καιρούς για να

ξεπεραστεί αυτό το εμπόδιο. Αυτές οι τεχνικές εφαρμόζονται με σκοπό την εξομάλυνση των N-grams πιθανοτήτων, έτσι ώστε κάθε N-gram που δεν έχει εμφανιστεί να μην έχει μηδενική πιθανότητα.

Για την εκτίμηση των παραπάνω πιθανοτήτων χρησιμοποιούνται μεγάλου μεγέθους κείμενα, τα οποία αποτελούν τα train data, όπως αναφέραμε και παραπάνω. Στη συνέχεια μετράμε τον αριθμό εμφανίσεων κάθε λέξης ή συνδυασμούς συνεχόμενων λέξεων ώστε να υπολογίσουμε τα unigrams, bigram, trigram, κλπ. Τα train data, για να έχει νόημα η όλη εκτίμηση του μοντέλου, πρέπει να προέρχονται από τον τομέα που θέλουμε να εκτιμήσουμε την ακρίβεια του μοντέλου ή να εφαρμόσουμε την αναγνώριση ομιλίας. Ένα πρότυπο έργο για την αγγλική γλώσσα αποτελούν τα Wall Street Journal corpus τα οποία θα χρησιμοποιήσουμε κατά την υλοποίηση αυτής της διπλωματικής εργασίας.

Υποθέτοντας ένα bigram μοντέλο, εκτιμούμε τη πιθανότητα $p(w_i|w_{i-1})$, μετρώντας τον αριθμό εμφανίσεων του bigram w_{i-1}, w_i και κανονικοποιούμε ως εξής:

$$p(w_i|w_{i-1}) = \frac{c(w_{i-1},w_i)}{c(w_{i-1})} = \frac{c(w_{i-1},w_i)}{\sum_{w_i} c(w_{i-1},w_i)}$$

Η παραπάνω σχέση ονομάζεται εκτίμηση μέγιστη πιθανοφάνειας(Maximum Likelihood-ML), διότι μεγιστοποιεί την πιθανότητα εμφάνισης των δεδομένων εκπαίδευσης. Είναι κατανοητό ότι αν ένα bigram δεν εμφανίζεται στα train data η πιθανότητα του θα είναι μηδέν. Αυτό αποτελεί ένα πολύ ανεπιθύμητο γεγονός, καθώς έχει σαν αποτέλεσμα σε μια «αποδεκτή» συμβολοσειρά να επιτρεπόταν να εκχωρηθεί μια μηδενική πιθανότητα. Για να ξεπεραστεί αυτό το πρόβλημα, θα πρέπει να εξομαλυνθούν οι κατανομές που λαμβάνονται από την εκτίμηση της Maximum Likelihood. Η εξομάλυνση αποσκοπεί στη κανονικοποίηση των πιθανοτήτων, ώστε να μην παίρνουν μηδενικές τιμές. Κάνοντας smoothing τις πιθανότητες δεν επηρεάζονται μόνο οι μηδενικές πιθανότητες, αλλά και οι πολύ μικρές πιθανότητες, όπως για παράδειγμα, η πιθανότητα των bigrams που λαμβάνουν χώρα μόνο μια φορά ή γενικά μικρό αριθμό φορών σε σχέση με το μέγεθος των train data, τα λεγόμενα και ως *singleton* in the training data

Υπάρχουν πολλές τεχνικές εξομάλυνσης που χρησιμοποιούνται ώστε να ξεπεράσουμε το πρόβλημα των μηδενικών πιθανοτήτων και να βελτιστοποιήσουμε τα N-grams μοντέλα. Μερικές από τις πιο γνωστές τεχνικές, είναι η Good-Turing Discounting, Absolute Discounting, Written-Bell Discounting, Kneser-Ney Discounting, Add-one smoothing και Katz's backing-off. Η περιγραφή τους και τη χρήση τους στην εξομάλυνση είναι ευρύτερα διαδεδομένη στη βιβλιογραφία, επομένως δεν θα επεκταθούμε σε αυτό το θέμα.

2.2 Θεωρία Πληροφορίας και Ποιότητα Γλωσσικών Μοντέλων

Τα γλωσσικά μοντέλα χρησιμοποιούνται για την εκτίμηση της πιθανότητας κάποιας ακολουθίας λέξεων. Για μια ακολουθία N λέξεων, ο όρος P(**W**) περιέχει πληροφορίες για την πιθανότητα της ακολουθίας καθώς και για την ακρίβειά της. Μπορούμε να αποφανθούμε για την ποιότητα του μοντέλου βασιζόμενοι στη πιθανότητα P(**W**) κάθε ακολουθίας δεδομένων. Η Εντροπία (Entropy) και το Perplexity αποτελούν δυο βασικές μονάδες μέτρησης για τον τομέα τη Θεωρία της Πληροφορίας και χρησιμοποιούνται για την αξιολόγηση της ποιότητας ενός γλωσσικού μοντέλου.

Η γλώσσα μπορεί να θεωρηθεί ως πηγή πληροφοριών των οποίων οι έξοδοι είναι λέξεις w_i οι οποίες ανήκουν στο λεξιλόγιο της γλώσσας αυτής. Η πιο κοινή μονάδα μέτρησης για την αξιολόγηση ενός γλωσσικού μοντέλου είναι ο ρυθμός σφάλματος (error rate) στην αναγνώριση λέξεων, το οποίο απαιτεί την συμμετοχή ενός συστήματος αναγνώρισης ομιλίας. Εναλλακτικά, μπορούμε να μετρήσουμε την πιθανότητα που το μοντέλο γλώσσας εκχωρεί στα test word strings χωρίς να είναι απαραίτητη η συμμετοχή των συστημάτων αναγνώρισης ομιλίας. Αυτό αποτελεί παράγωγο μέτρο του cross-entropy, γνωστό και ως test-set perplexity.

2.2.1 Εντροπία

Δεδομένου ενός γλωσσικού μοντέλου που αναθέτει την πιθανότητα $P(\mathbf{W})$ σε μια ακολουθία λέξεων **W**, μπορούμε να παράγουμε έναν αλγόριθμο συμπίεσης που κωδικοποιεί το κείμενο **W** χρησιμοποιώντας τα $-log_2 P(\mathbf{W})$ bits. Το cross-entropy $H(\mathbf{W})$ για ένα μοντέλο $P(w_i|w_{i-n+1}, ..., w_{i-1})$ στα δεδομένα **W**, για μια αρκετά μεγάλη ακολουθία λέξεων, μπορεί να προσδιοριστεί ως:

$$H(\mathbf{W}) = -\frac{1}{N_W} log_2 P(\mathbf{W})$$

όπου N_W είναι το μέγεθος του κειμένου **W** υπολογισμένο σε λέξεις.

2.2.2 Perplexity

Το perplexity $PP(\mathbf{W})$ ενός γλωσσικού μοντέλου $P(\mathbf{W})$ ορίζεται ως το αντίστροφο της (γεωμετρικής) μέσης πιθανότητας που έχει ανατεθεί από το μοντέλο στο test set **W**. Η

έννοια αυτή, σχετίζεται με το cross-entropy, και συνήθως αναφαίρετε ως test set perplexity:

$$PP(\mathbf{W}) = 2^{H(\mathbf{W})}$$

Το perplexity μπορεί γενικά να ερμηνευθεί ως ο γεωμετρικός μέσος του παράγοντα διακλάδωσης του κειμένου όταν παρουσιάζεται στο γλωσσικό μοντέλο. Το perplexity καθορίζεται από δύο βασικές παραμέτρους : ένα γλωσσικό μοντέλο και μια ακολουθία λέξεων. Το test-set perplexity αξιολογεί την ικανότητα γενίκευσης του μοντέλου γλώσσας. Από την άλλη, το train-set perplexity εκτιμά κατά πόσο το γλωσσικό μοντέλο ταιριάζει με τα δεδομένα εκπαίδευσης. Η γενικότερη ιδέα, βασίζεται στο γεγονός ότι οι χαμηλότερες τιμές του perplexity συσχετίζονται με την καλύτερη μοντελοποίηση γλώσσας. Επομένως, στόχος ήταν και είναι η επίτευξη όσο το δυνατόν χαμηλότερου perplexity. Αυτό συμβαίνει επειδή το perplexity είναι ουσιαστικά ένα στατιστικώς σταθμισμένο μέτρο διακλάδωσης λέξης για το test set.

Για κάθε γλωσσικό μοντέλο, υπάρχει η δυνατότητα να υπολογίσουμε το perplexity για το test set του. Η ελάχιστη τιμή που μπορεί να πάρει το perplexity είναι ένα, κάτι που σημαίνει ότι στη περίπτωση αυτή όλες οι λέξεις μιας ακολουθίας έχουν πιθανότητα ίση προς ένα. Από την άλλη πλευρά, αν μια λέξη έχει μηδενική πιθανότητα, τότε η πιθανότητα ολόκληρης της πρότασης θα είναι μηδέν και η τιμή του perplexity θα είναι άπειρη. Έτσι, μπορούμε να υποθέσουμε ότι η πρόκληση στο γλωσσικό μοντέλο είναι να αποφευχθούν οι μηδενικές πιθανότητες. Ένα καλοφτιαγμένο μοντέλο θα πρέπει να δίνει όσο το δυνατόν μικρότερο perplexity για μεγάλα σύνολα test data. *Η τιμή του perplexity είναι ένα μέτρο σύγκρισης, της ποιότητας διαφορετικών γλωσσικών μοντέλων, για κοινά test data.*

2.3 Ομαδοποίηση

2.3.1 Ομαδοποίηση Δεδομένων: Βασικές Έννοιες

Ομαδοποίηση δεδομένων είναι η ταξινόμηση προτύπων όπου δεν υπάρχει επίβλεψη (unsupervised classification), δηλαδή η περίπτωση όπου η κλάση στην οποία ανήκει κάθε πρότυπο εκπαίδευσης (training pattern) δεν είναι γνωστή. Έτσι, η κύρια μέριμνά μας είναι οι «λογικές» ομάδες (clusters ή groups), πράγμα το οποίο θα μας επιτρέψει να ανακαλύψουμε χρήσιμες ομοιότητες και διαφορές μεταξύ των προτύπων και να εξάγουμε χρήσιμα συμπεράσματα σχετικά με αυτά. Η διαδικασία αυτή της ανάδειξης ομάδων (ομαδοποίηση, clustering),

που σχηματίζονται από τα πρότυπα, συναντάται σε πολλά επιστημονικά πεδία, όπως οι επιστήμες της ζωής (βιολογία, ζωολογία), οι ιατρικές επιστήμες (ψυχιατρική, παθολογία), οι κοινωνικές επιστήμες (κοινωνιολογία, αρχαιολογία), οι επιστήμες της Γης (γεωγραφία, γεωλογία) και των τεχνολογιών (engineering). Η ομαδοποίηση δεδομένων (clustering) απαντάται με διαφορετικά ονόματα σε διαφορετικά πεδία εφαρμογών. Έτσι, συναντάται ως εκμάθηση χωρίς επίβλεψη ή εκμάθηση χωρίς διδάσκαλο (στην αναγνώριση προτύπων), αριθμητική ταξινομία (στην βιολογία και την οικολογία), τυπολογία (στις κοινωνικές επιστήμες) και διαμέριση (στη θεωρία γράφων).

Η διαδικασία κατανομής των δεδομένων σε ομάδες βασίζεται σε συγκεκριμένα κριτήρια ομαδοποίησης. Διαφορετικά κριτήρια μπορεί να έχουν διαφορετικά αποτελέσματα ομαδοποίησης.

Τα βήματα που πρέπει να ακολουθήσουμε προκειμένου να οργανώσουμε μια διαδικασία ομαδοποίησης, είναι τα ακόλουθα (Pattern Recognition Book [4]):

- *Επιλογή χαρακτηριστικών (feature selection).* Τα χαρακτηριστικά πρέπει να επιλεγούν κατάλληλα έτσι ώστε να κωδικοποιούν όσο το δυνατόν περισσότερη πληροφορία σχετικά με το υπό εξέταση πρόβλημα. Επιπλέον, ο ελάχιστος δυνατός πλεονασμός πληροφορίας μεταξύ των χαρακτηριστικών είναι ένα σημαντικό ζητούμενο και είναι πολύ πιθανό να είναι αναγκαία η προ-επεξεργασία των χαρακτηριστικών πριν την χρησιμοποίηση τους στα επόμενα στάδια.

- *Μέτρο εγγύτητας (proximity measure).* Το μέτρο αυτό ποσοτικοποιεί το βαθμό «ομοιότητας» ή «ανομοιότητας» μεταξύ δυο διανυσματικών χαρακτηριστικών. Είναι φυσικό να εξασφαλίσουμε ότι όλα τα επιλεγμένα χαρακτηριστικά συνεισφέρουν εξίσου στον υπολογισμό του μέτρου εγγύτητας και ότι δεν υπάρχουν κάποια χαρακτηριστικά που κυριαρχούν έναντι των άλλων. Για το θέμα αυτό πρέπει να ληφθεί μέριμνα κατά το στάδιο της προ-επεξεργασίας των χαρακτηριστικών.

- *Κριτήριο ομαδοποίησης (clustering criterion).* Η επιλογή του κριτηρίου αυτού εξαρτάται από την ερμηνεία που δίνουμε στον όρο «λογικές ομάδες», βασιζόμενος στον τύπο/μορφή των ομάδων που, κατά την εκτίμησή του, σχηματίζουν τα διανύσματα του συνόλου των δεδομένων. Το κριτήριο ομαδοποίησης μπορεί να εκφραστεί μέσω μιας συνάρτησης κόστους ή μέσω άλλων τύπων κανόνων.

- *Αλγόριθμοι ομαδοποίησης (clsutering algorithms).* Έχοντας υιοθετήσει ένα μέτρο εγγύτητας και ένα κριτήριο ομαδοποίησης, το παρόν βήμα αναφέρεται στην επιλογή

ενός αλγοριθμικού σχήματος, το οποίο θα ανιχνεύσει την δομή της ομαδοποίησης (clustering structure) του συνόλου δεδομένων (δηλαδή τις ομάδες που σχηματίζονται από τα διανύσματα χαρακτηριστικών του συνόλου δεδομένων).

- *Επικύρωση των αποτελεσμάτων (validation of results).* Έχοντας διαθέσιμα τα αποτελέσματα του αλγορίθμου ομαδοποίησης, θα πρέπει να τα εξετάσουμε ως προς την ορθότητα τους. Αυτό γίνεται, συνήθως με τη χρήση κατάλληλων δοκιμών (tests).

- *Ερμηνεία των αποτελεσμάτων (interpretation of the results).* Σε πολλές περιπτώσεις, στο πεδίο εφαρμογής στο οποίο εμπίπτει το υπό εξέταση πρόβλημα, θα πρέπει να συνεκτιμήσουμε τα αποτελέσματα της ομαδοποίησης μαζί με άλλα πειραματικά στοιχεία προκειμένου να εξαχθούν ορθά συμπεράσματα.

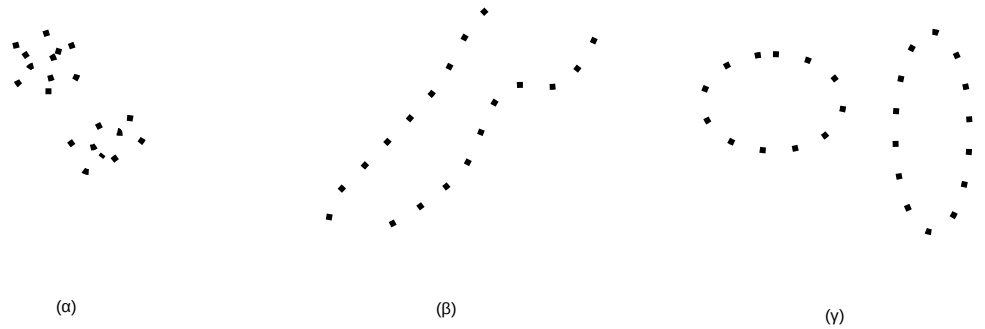
Σε κάποιες περιπτώσεις θα πρέπει να περιλαμβάνουμε και ένα επιπλέον βήμα, το οποίο είναι γνωστό ως *τάση ομαδοποίησης (clustering tendency)*. Το βήμα αυτό αναφέρεται στη χρήση διαφόρων δοκιμών (tests), τα οποία υποδεικνύουν την ύπαρξη ή όχι δομής ομαδοποίησης στο υπό εξέταση σύνολο δεδομένων. Για παράδειγμα, αν ένα σύνολο δεδομένων έχει εντελώς τυχαία δομή η αναζήτηση δομής ομαδοποίησης σε αυτό δεν έχει νόημα.

Όπως έχει ήδη υπονοηθεί, διαφορετικές επιλογές χαρακτηριστικών, μέτρου εγγύτητας, κριτηρίου ομαδοποίησης και αλγορίθμου ομαδοποίησης μπορούν να οδηγήσουν σε εντελώς διαφορετικές ομαδοποιήσεις δεδομένων. Με άλλα λόγια, *η υποκειμενικότητα είναι μια πραγματικότητα με την οποία θα πρέπει να ζήσουμε από εδώ και πέρα*.

2.3.2 Ορισμός Ομαδοποίησης Δεδομένων

Ο ορισμός της ομαδοποίησης δεδομένων (Pattern Recognition Book [4]) οδηγεί κατ' ευθείαν στον ορισμό της ομάδας. Πολλοί ορισμοί έχουν προταθεί με την πάροδο των ετών. Ωστόσο, οι περισσότεροι από αυτούς βασίζονται σε όρους που δεν είναι σαφώς ορισμένοι, όπως «παρόμοια», «ομοειδή» κλπ. ή είναι προσανατολισμένοι σε ένα συγκεκριμένο είδος ομάδας. Οι περισσότεροι από αυτούς τους ορισμούς είναι ασαφείς και παρουσιάζουν το φαινόμενο της κυκλικής αναφοράς. Το γεγονός αυτό καταδεικνύει τη δυσκολία διατύπωσης ενός καθολικά αποδεκτού ορισμού για τον όρο ομάδα.

Υποθέτουμε ότι τα διανύσματα εκλαμβάνονται ως σημεία του l-διάστατου χώρου και οι ομάδες περιγράφονται ως «συνεχείς περιοχές του χώρου αυτού, όπου η πυκνότητα σημείων είναι σχετικά υψηλή, και οι οποίες χωρίζονται μεταξύ τους από περιοχές οι οποίες χαρακτηρίζονται από σχετικά χαμηλή πυκνότητα σε σημεία». Ο ορισμός αυτός είναι εγγύτερα στην



Σχήμα 2.2: (α)Συμπαγές ομάδες. (β) Επιμήκεις ομάδες. (γ) Σφαιρικές και ελλειψοειδείς ομάδες.

οπτική αντίληψη που έχουμε για τις ομάδες σε χώρους δυο ή τριών διαστάσεων, ώστε να είναι όσο περισσότερο κατανοητός γίνεται. Έστω X το υπό εξέταση σύνολο δεδομένων, δηλαδή:

$$X = \{x_1, x_2, ..., x_N\}$$

Ορίζουμε ως m-ομαδοποίηση του X, $\mathfrak{R}$, το διαμερισμό του X σε m σύνολα (ομάδες), $C1, ..., C_m$, έτσι ώστε να ικανοποιούνται οι ακόλουθες τρεις συνθήκες:

1. $C_i \neq \emptyset, i = 1, ..., m$
2. $\bigcup_{i=1}^{m} C_i = X$
3. $C_i \bigcap C_j = \emptyset, i \neq j, i, j = 1, ...m$

Επιπροσθέτως, τα διανύσματα που περιέχονται στην ομάδα C_i, είναι «πιο όμοια» μεταξύ τους και «λιγότερο όμοια» με διανύσματα των άλλων ομάδων. Η ποσοτικοποίηση των όρων «όμοια» και «ανόμοια» εξαρτάται σε μεγάλο βαθμό από τον τύπο των εμπλεκόμενων ομάδων. Για παράδειγμα, άλλα μέτρα (που μετρούν την ομοιότητα) είναι κατάλληλα για τη μέτρηση της ομοιότητας όταν έχουμε συμπαγείς ομάδες (π.χ. 2.2.α), άλλα μέτρα είναι κατάλληλα όταν έχουμε επιμήκεις ομάδες (π.χ. 2.2.β) και άλλα όταν έχουμε ομάδες με κελυφωτό σχήμα (shell-shaped) (π.χ 2.2.γ)

2.4 Ιστορική Αναδρομή

Η συστηματική μελέτη για τη δημιουργία και την εφαρμογή στατιστικών γλωσσικών μοντέλων πάνω σε κείμενα άρχισε περίπου το 1995 και το ενδιαφέρον παραμένει μεγάλο μέχρι και σήμερα. Σχεδόν όλη η δουλειά που είχε παρουσιαστεί τα πρώτα χρόνια, είχε αναπτυχθεί πάνω στο διακριτό χώρο. Πληθώρα εργαλείων έχουν εμφανιστεί κατά καιρούς για το σκοπό αυτό και υπάρχουν διαθέσιμα στο διαδίκτυο. Ένα από τα πρώτα και πιο διαδεδομένα εργαλεία ήταν το SRILM (An Extensible Language Modeling Toolkit) το οποίο θα χρησιμοποιήσουμε για την εξαγωγή ενός πρώτου baseline πάνω τα δεδομένα μας, στο διακριτό χρόνο, πριν μοντελοποιήσουμε την όλη διαδικασία στο συνεχές χώρο. Άλλα παρόμοια σύνολα εργαλείων για την εκτίμηση και την αναπαράσταση γλωσσικών μοντέλων είναι:

- **IRSLM** όπου βρίσκουμε ενδεικτικές δημοσιεύσεις από τους Marcello Federico and Nicola Bertoldi [5], [6] etc.
- **MITLM**. όπου βρίσκουμε ενδεικτικές δημοσιεύσεις από τον Bo-June (Paul) Hsu and James Glass [7], [8] etc.

Όπως έχουμε ήδη τονίσει, η ιδέα για γλωσσικά μοντέλα στο συνεχή χρόνο δεν είναι εντελώς καινούργια. Αυτή η νέα προοπτική, έχει αναφερθεί από τους Affify et al. [1] και [9], Chen [2] και Schwenk-Gauvain [10] κάποια χρόνια πριν και θα αποτελέσει τα βασικά θεμέλια της εργασίας μας. Η ερευνητική διαδικασία που θα μας απασχολήσει εκτενέστερα και είναι μια από τις τελευταίες προσεγγίσεις, είναι η διπλωματική εργασία του Τσιάκα [3] που θα αποτελέσει και τη βάση σύγκρισης (baseline) με τη παρούσα διπλωματική εργασία. Αναλυτικότερα, στη συγκεκριμένη εργασία, κατασκευάστηκαν γλωσσικά μοντέλα με χρήση συνεχών κατανομών και παρεμβολή N-grams μοντέλων για τον υπολογισμό της πιθανότητα των λέξεων, όπου η είσοδος των δεδομένων γινόταν λαμβάνοντας υπόψη κάθε λέξη ξεχωριστά και χρησιμοποιώντας ένα περιορισμένο σε μέγεθος λεξιλόγιο.

Η ακριβής υλοποίηση της δουλειάς μας, δηλαδή η χρήση μεγάλου σε μέγεθος λεξιλογίου με ταυτόχρονη εφαρμογή διαφόρων τεχνικών clustering στα δεδομένα εισόδου, με σκοπό την δημιουργία μοντέλων με μεγαλύτερη ακρίβεια, είναι εφαρμογές πάνω στις οποίες δεν υπάρχουν κάποιες αντίστοιχες δουλειές. Επομένως, στόχος μας θα είναι να βελτιώσουμε τα αποτελέσματα που αφορούν το απλουστευμένο μοντέλο που υλοποιήθηκε από τον Τσιάκα και να πειραματιστούμε στην ακρίβεια του μοντέλου όταν γίνεται χρήση μεγάλου λεξιλογίου.

Κεφάλαιο 3

Γλωσσικά Μοντέλα στο Συνεχή Χώρο

3.1 Εισαγωγή

Όπως αναλύσαμε στα προηγούμενα κεφάλαια, τα N-grams μοντέλα αποτελούν κυρίαρχη τεχνολογία στην μοντελοποίηση γλώσσας. Παρά την ευρύτερα διαδεδομένη εφαρμογή τους, τα διακριτά N-grams μοντέλα «πάσχουν» από δύο βασικά μειονεκτήματα. Μπορούμε να αναφερόμαστε σε αυτά ως *γενίκευση (generalization)* και ως *προσαρμοστικότητα (adaptability)*. Τα δύο αυτά προβλήματα, αναφέρονται στα N- grams με μηδενική πιθανότητα και στις παραμέτρους του N-gram μοντέλου. Συνήθως, ένα N-gram μοντέλο έχει τεράστιο αριθμό από παραμέτρους. Αυτό έχει σαν αποτέλεσμα, να είναι πολύ δύσκολο να προσαρμοστεί όταν χρησιμοποιείται ένα σχετικά μικρό ποσό από δεδομένα. Προτείνουμε, γλωσσικά μοντέλα στο συνεχή χώρο, προκειμένου να ξεπεραστούν αυτά τα προβλήματα και να εκμεταλλευτούμε τα σημαντικά πλεονεκτήματα τους, όπως η εξομάλυνση (smoothing) σε unseen γεγονότα, και η ευκολία προσαρμογής του μοντέλου (model adaptation). Η βασική ιδέα αυτών των μοντέλων είναι η ομοιότητα των λέξεων ή κλάσεων από λέξεις, όπως αυτά προβάλλονται σε έναν συνεχή παραμετρικό χώρο. Δηλαδή, μερικές λέξεις/κλάσεις ή N-grams είναι «πιο κοντά» μεταξύ τους από άλλες λέξεις/κλάσεις ή N-grams αντίστοιχα.

Υπάρχουν ορισμένα σημαντικά ζητήματα σχετικά με αυτά τα μοντέλα. Αρχικά, θα πρέπει να γίνει η κατάλληλη προβολή της κάθε «διακριτικής» κλάσης στο νέο συνεχή χώρο. Στη συνέχεια, θα πρέπει να ασχοληθούμε με τους χώρους μεγάλων διαστάσεων που προκύπτουν. Η μοντελοποίηση σε μεγάλες διαστάσεις, αναφέρεται και ως curse of dimensonality (κατάρα

των διαστάσεων), περιέχει πολλές δυσκολίες. Επομένως, στοχεύουμε στην κατασκευή ενός γλωσσικού μοντέλου στο συνεχή χώρο το οποίο περιλαμβάνει: αντιστοίχηση κλάσεις λέξεων του διακριτού χώρου σε μια συνεχή αναπαράσταση (class mapping), και ένα ταξινομητή (classifier) που αποφασίζει την επόμενη κλάση-λέξη με βάση τα mapped ιστορικά στο συνεχή χώρο που έχουν προκύψει, ή ισοδύναμα εκχωρεί μια πιθανότητα σε κάθε λέξη δεδομένης της ιστορίας της και της κλάσεις στην οποία ανήκει.

3.2 Αντιστοίχιση κλάσεων σε διανύσματα

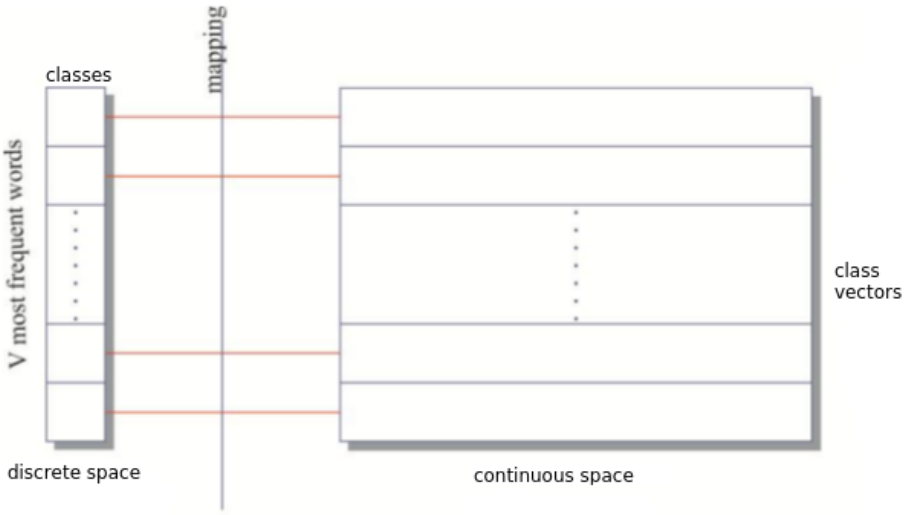
Class mapping είναι η προβολή κάθε κλάσης λέξεων από τα δεδομένα μας στο νέο συνεχή χώρο. Όσον αφορά το μεγάλο αριθμό λέξεων καθώς και την δυσκολία στην επεξεργασία των παραμέτρων σε χώρο μεγάλων διαστάσεων, θα δουλέψουμε πάνω σε δυο διαφορετικές προσεγγίσεις:

- Η πρώτη προσέγγιση αφορά την προσπάθεια ν΄ αναπαραστήσουμε τις πιο συχνές λέξεις από τα train data. Πιο συγκεκριμένα, επιλέγουμε τις V πιο συχνές λέξεις, οι οποίες αποτελούν το μοντέλο του λεξιλογίου μας και είναι οι λέξεις στις οποίες θα εφαρμοστεί το word clustering. Επίσης, καθορίζουμε μια κλάση *unk* η οποία αντιπροσωπεύει τις λέξεις οι οποίες δεν συμπεριλαμβάνονται στο λεξιλόγιο. Για τον προσδιορισμό του class mapping, θα πρέπει να ληφθεί υπόψη η συχνότητα και η σημασία της λέξης/κλάσης, καθώς επίσης και η σχέση κάθε κλάσης με τις άλλες κλάσεις που σχηματίζονται (3.1).

- Η δεύτερη προσέγγιση άφορα την αναπαράσταση όλων των διαφορετικών λέξεων που εμφανίζονται στα train data. Αναλυτικότερα, το σύνολο όλων των διαφορετικών λέξεων που υπάρχουν στα δεδομένα μας, θα αποτελέσουν και το μοντέλο του λεξιλογίου μας, που ομοίως με πριν θα εφαρμοστεί το word clustering. Προφανώς, στη περίπτωση αυτή δεν υπάρχει κλάση *unk*, υπάρχουν όμως λέξεις στα test data οι οποίες δεν εμφανίζονται στα train data και τις οποίες για λόγους που θα εξηγήσουμε παρακάτω θα πρέπει επίσης να εφαρμόσουμε clustering.

3.2.1 Διανύσματα χαρακτηριστικών των κλάσεων

Κάθε κλάση λέξεων μπορεί να αναπαρασταθεί από έναν δείκτη σε ένα διάνυσμα (indicator vectors), c_i, έχοντας ένα στη i^{th} θέση και μηδενικά στις υπόλοιπες K-1 θέσεις (όπου K ο



Σχήμα 3.1: Class mapping

αριθμός των κλάσεων). Έτσι, έχουμε έναν KxK πίνακα, που περιέχει όλα τα διανύσματα κλάσεων. Αυτό ο πίνακας χαρακτηριστικών διανυσμάτων, είναι αραιός (sparse) και έχει την μορφή που περιγράφεται στο πίνακα 3.1.

c_1	1	0	0	0	0	...	0
c_2	0	1	0	0	0	...	0
c_3	0	0	1	0	0	...	0
c_4	0	0	0	1	0	...	0
c_5	0	0	0	0	1	...	0
...	...	...	...	...	...	...	...
c_K	0	0	0	0	0	...	1

Πίνακας 3.1: Indicator Vectors

Ο παραπάνω πίνακας αποτελείται από K^2 στοιχεία. Αυτό σημαίνει ότι, καθώς ο αριθμός των κλάσεων αυξάνεται έχει σαν αποτέλεσμα το μέγεθος του πίνακα να αυξάνεται εκθετικά. Το επόμενο βήμα είναι η αντιστοίχηση κάθε διανύσματος σε χαμηλότερη διάσταση, ανάλογα με τη συχνότητα της κλάσης-λέξης, και προσδιορισμός στον τρόπο με τον οποίο οι λέξεις μιας συγκεκριμένης κλάσης συμπεριφέρονται με τις λέξεις που ανήκουν σε διαφορετικές κλάσεις.

C_{ij}	c_1	c_2	c_3	c_4	c_5	...	c_K
c_1	5	3	127	405	16	...	65
c_2	2	1	0	0	0	...	0
c_3	0	3	0	2	0	...	1
c_4	0	1	0	3	0	...	2
c_5	1	0	0	0	0	...	0
...	...	...	...	...	...	...	...
c_K	1	1	1	5	0	...	6

Πίνακας 3.2: Class Co-occurence Matrix

3.2.2 Class Co-occurences

Όπως αναφέραμε και παραπάνω, για να γίνει το κατάλληλο class mapping, παρατηρούμε πως σχετίζεται κάθε κλάση με τις υπόλοιπες. Πιο συγκεκριμένα, υπολογίζουμε τον πίνακα συν-εμφανίσεων κλάσεων (class co-occurences) C_{ij}, που απεικονίζει τη σχέση μεταξύ των κλάσεων υπολογίζοντας το bigram για κάθε λέξη i kai j αντλώντας σαν πληροφορία την κλάση στην οποία ανήκουν η i και j λέξη. Κάθε στοιχείο c_{ij} είναι ο αριθμός των φορών που μια λέξη της κλάσης i ακολουθείται από μια λέξη της κλάσης j, στα δεδομένα εκπαίδευσης.

Ο class co-oocurence matrix (πίνακας 3.2) είναι ένας KxK πίνακας που περιέχει πολλά μηδενικά στοιχεία, καθώς υπάρχουν πολλές λέξεις διαφόρων κλάσεων που δεν εμφανίζονται «μαζί», δηλαδή δεν δημιουργούν κάποιο bigram, στα δεδομένα εκπαίδευσης. Ένας άλλος λόγος της ύπαρξης των μηδενικών είναι ότι σε πολλές περιπτώσεις υπάρχουν κλάσεις με πολύ μικρό αριθμό λέξεων-μελών ή κλάσεις που περιέχουν λέξεις οι οποίες έχουν πολύ μικρή συχνότητα εμφάνιση με αποτέλεσμα να μην δημιουργούνται από τις κλάσεις αυτές πολλά ζευγάρια λέξεων (bigram). Υπάρχουν και κάποιες πιο σπάνιες περιπτώσεις, όπου ο αριθμός των κλάσεων είναι αρκετά μικρός με αποτέλεσμα ο class co-occurence matrix να μην εμφανίζει πολλά μηδενικά όντας αρκετά πυκνός(dense). Σε αυτή τη περίπτωση όμως, εμφανίζονται άλλα προβλήματα αποφεύγοντας έτσι την επιλογή μικρού αριθμού κλάσεων, όπως θα αναλύσουμε παρακάτω.

Κάθε γραμμή αναπαριστά κάθε διάνυσμα κλάσης (class vector) και οι στήλες αντιπροσωπεύουν τα ιστορικά. Για την αντιμετώπιση των μηδενικών στοιχείων, γίνεται smoothing σύμφωνα με τη σχέση $c_{ij} = log(1+c_{ij})$. Όσον αφορά το μεγάλο αριθμό των στοιχείων, χρησιμοποιούμε διάφορες τεχνικές μείωσης διαστάσεων. Σε πρώτη φάση, εφαρμόζουμε στον

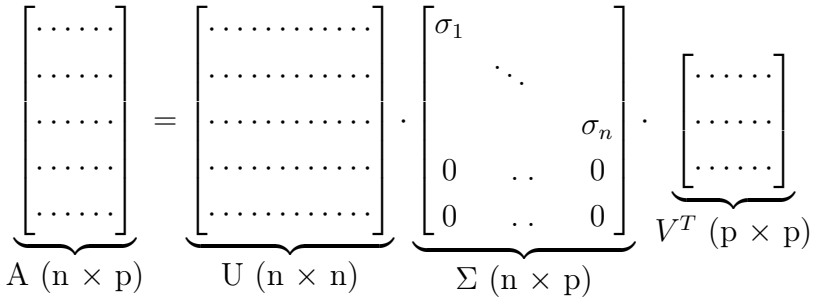


Figure 3.2: Σχηματική αναπαράσταση της τεχνικής SVD

class co-oocurence matrix μια τεχνική συμπίεσης δεδομένων, η οποία ονομάζεται Ανάλυση στη Βάση των Ιδιάζουσων Τιμών (Singular Value Decomposition-SVD).

3.2.3 Singular Value Decomposition (SVD)

Ο SVD είναι μια από τις περισσότερο κομψές και ισχυρές μεθόδους γραμμικής άλγεβρας, και έχει χρησιμοποιηθεί εκτενώς για την μείωση του βαθμού και της διάστασης σε προβλήματα αναγνώρισης προτύπων και εφαρμογές ανάκτησης πληροφορίας. Αναλυτικότερα, η τεχνική αυτή παραγοντοποιεί έναν πίνακα $A \in R^{m \times n}$ διαστάσεων $n \times p$ σε ένα γινόμενο τριών επιμέρους πινάκων: έναν πίνακα U διαστάσεων $n \times n$, έναν πίνακα Σ διαστάσεων $n \times p$ και έναν πίνακα V^T διαστάσεων $p \times p$. Ο ορισμός λοιπόν έχει ως εξής:

$$A_{n \times p} = U_{n \times n} * \Sigma_{n \times p} * V_{p \times p}$$

Οι U και V^T είναι ορθογώνιοι πίνακες (δηλαδή ισχύει $U^T \cdot U = I_{n \times n}$ και $V^T \cdot V = I_{p \times p}$), με τις n στήλες του U και τις p στήλες του V (ή τις γραμμές του V^T), να ονομάζονται left και right singular vectors του A και να αποτελούν τα ιδιοδιανύσματα του $A \cdot A^T$ και του $A^T \cdot A$ αντίστοιχα. Ο Σ από την άλλη,είναι ένας τετραγωνικός διαγώνιος πίνακας με μη αρνητικές πραγματικές τιμές ταξινομημένες σε φθίνουσα διάταξη ($\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$). Οι (διαγώνιες) τιμές του $\sigma_1, \sigma_2, \cdots, \sigma_n$, είναι γνωστές ως singular values του πίνακα A και αποτελούν την τετραγωνική ρίζα των ιδιοτιμών του $A \cdot A^T$ και του $A^T \cdot A$. Για να υπολογίσουμε τον SVD πρέπει να βρούμε τις ιδιοτιμές (eigenvalues) και τα ιδιοδιανύσματα (eigenvectors), τα οποία μας καταδεικνύουν ποια features περιέχουν περισσότερη πληροφορία και ποια μπορούν να απαλειφθούν.

Πιο αναλυτικά, οι singular values πολλαπλασιάζονται μόνο με ορισμένες στήλες του U και του V. Επομένως, είναι αυτές που καθορίζουν τον τρόπο με τον οποίο οι στήλες του U και V επηρεάζουν τον πίνακα. Εάν η τιμή είναι αρκετά μικρή τότε η στήλη ή η γραμμή της δεν προστίθεται στο νέο πίνακα. Πολλές από τις τιμές είναι μηδέν. Αυτό σημαίνει ότι οι αντίστοιχες στήλες και γραμμές που δεν θα έχουν χρήσιμες πληροφορίες. Έτσι, επιχειρούμε να μετασχηματίσουμε το πίνακα, έτσι ώστε περιέχει τις πολύτιμες πληροφορίες.

Συχνά, οι singular values κατατάσσονται σύμφωνα με τη σημασία των γραμμών και των στηλών τους. Αυτός είναι ο λόγος που ο SVD χρησιμοποιείται για την αποσύνθεση πινάκων και τη μείωση των δεδομένων. Αν θέλουμε να διατηρήσουμε τις βασικές πληροφορίες ενός πίνακα με τη μείωση της διάστασης, μπορούμε να κρατήσουμε τα M μεγαλύτερες ιδιάζουσες τιμές και να κατασκευάσουμε έναν άλλον πίνακα Σ′, που περιέχει μόνο αυτές M singular values. Έτσι, μπορούμε να πάρουμε ένα μετασχηματισμένο πίνακα $M \times N$.

Χρησιμοποιώντας τις x μεγαλύτερες singular values, μπορούμε να προβάλλουμε κάθε class vector στον M-διάστατο χώρο, για M = x. Προβάλλουμε κάθε class vectors c_i στο u_i χρησιμοποιώντας την προβολή $u_i = A \cdot C$, όπου A είναι η έξοδος του SVD του co-occurence matrix για τις M μεγαλύτερες singular values. Ειδικότερα, ο πίνακας A σχηματίζεται από τα αριστερά singular vectors του SVD, δηλαδή τον πίνακα U (ο πίνακας που χρησιμοποιήθηκε και στην δική μας υλοποίηση). Αν υποθέσουμε ότι η έξοδος της αποσύνθεσης είναι $[U, \Sigma, V]$, τότε η προβολή της κλάσης είναι $u_i = U' \cdot C$.

Ο SVD εξυπηρετεί όταν υπάρχουν στα δεδομένα μας πλεονάζοντα features, αλλά δε βοηθά σε όλες τις περιπτώσεις αναφορικά με το sparcity των δεδομένων: δύο features μπορούν να είναι sparse, αλλά να εμπεριέχουν χρήσιμη πληροφορία που εξυπηρετεί την ταξινόμηση, οπότε δε μπορούμε να αφαιρέσουμε κανένα από τα δύο. Επιπλέον, η χρησιμοποίηση μιας μικρότερης διάστασης, δε μας εγγυάται πάντα πως θα έχουμε και την καλύτερη δυνατή απόδοση. Αυτό είχε σαν αποτέλεσμα, την εκτέλεση μεγάλου αριθμού πειραμάτων ώστε να καταλήξουμε στην πιο αποδοτική διάσταση.

3.3 History mapping

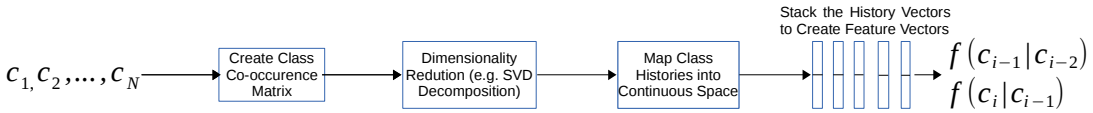
Με βάση τα N-grams μοντέλα, το ιστορικό κάθε κλάσης αποτελείται από τις προηγούμενες N-1 κλάσεις. Η επιλογή της κατάλληλης τιμής του N εξαρτάται από την εφαρμογή. Αν N=1, κάθε λέξη είναι ανεξάρτητη. Για N=2, κάθε λέξη εξαρτάται από την προηγούμενη και εάν N=3, κάθε λέξη εξαρτάται από τις προηγούμενες 2 λέξεις. Στη φυσική γλώσσα, έχει επικρατήσει η άποψη ότι κάθε λέξη έχει ισχυρή εξάρτηση από τις δύο προηγούμενές

της, έτσι αποφασίσαμε να εκπαιδεύσουμε trigram μοντέλα, για N=3.

"Whether the fashion designers who license their names will agree to stay on in the new company comma remains a key to the viability of the specific group period"

Παρατηρήσουμε από το παραπάνω παράδειγμα ότι η λέξη "designers", είναι στενά συνδεδεμένη με τις δύο προηγούμενες λέξεις, "The fashion designers". Επιπλέον, τα trigram "the new company" και "the specific group" είναι δύο ακόμη χαρακτηριστικά παραδείγματα που δείχνουν την εξάρτηση των λέξεων στο σχηματισμό των trigram. Να υπενθυμίσουμε και σε αυτό το σημείο, ότι στην υλοποίηση μας, δίνουμε έμφαση στις κλάσεις που ανήκουν αυτές οι λέξεις και όχι στις λέξεις αυτές καθ΄ αυτές.

Το επόμενο βήμα είναι να συγκεντρώσου όλα τα ιστορικά για κάθε κλάση. Αυτό υλοποιείται εντοπίζοντας για κάθε κλάση, τις N-1 προηγούμενες. Χρησιμοποιώντας την παραπάνω αντιστοίχιση mapping και λαμβάνοντας υπόψη τη κλάση κάθε λέξης, το ιστορικό κάθε κλάσης αποτελείται από τη συνένωση των κατάλληλων mapped κλάσεων. Κάθε ιστορικό είναι ένα διάνυσμα με $M \cdot (N-1)$ στοιχεία (3.3).



Σχήμα 3.3: History mapping

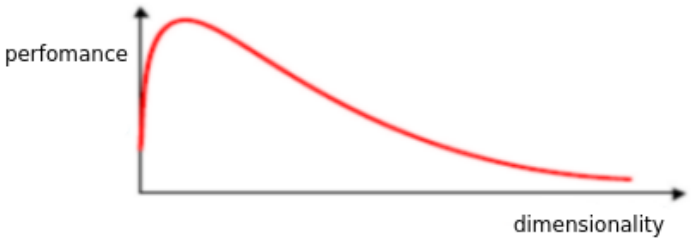
Έχοντας όλα τα διανύσματα ιστορικών, θα μπορούσαμε να μοντελοποιήσουμε τις παραμέτρους μας, με τη χρήση Gaussian κατανομών. Πρέπει να συλλέξουμε τα ιστορικά κάθε κλάσης από τα training data και στη συνέχεια να εκπαιδεύσουμε ένα μοντέλο για κάθε κλάση. Όπως έχει ήδη προαναφερθεί, η μοντελοποίηση είναι δύσκολη σε μεγάλες διαστάσεις. Ας υποθέσουμε ότι έχουμε ένα traigram μοντέλο (N=3), αριθμό κλάσεων C=1024 και πραγματοποιούμε SVD επιλέγοντας τις M=120 singular values. Αυτό σημαίνει ότι κάθε ιστορικό αποτελείται από $120 \times (N-1) = 240$ στοιχεία. Παρά τη χρήση του SVD, παραμένει ο μεγάλος χώρος διαστάσεων, με αποτέλεσμα να είναι ορατή η εμφάνιση της κατάρα της διαστατικότητας (Curse of Dimensionality). Μία ακόμη αποτελεσματική αντιστοίχηση για τα διανύσματα των ιστορικών είναι η $y_i = B \cdot h_i$ όπου y_i είναι η προβολή των ιστορικών διανυσμάτων h_i στο νέο-διάστατο χώρο. Η ακριβής εκτίμηση του μετασχηματισμού, του

πίνακα B περιγράφεται παρακάτω.

3.3.1 Η κατάρα της διαστατικότητας

Ο όρος, κατάρα της διαστατικότητας, εισήχθηκε το 1961 από τον Bellman και αναφέρεται στο πρόβλημα της ανάλυσης δεδομένων πολλών μεταβλητών καθώς αυξάνει η διάσταση.

Στην πραγματικότητα η κατάρα της διαστατικότητας σημαίνει ότι για δεδομένο αριθμό δειγμάτων, υπάρχει μια μέγιστη διάσταση των χαρακτηριστικών διανυσμάτων πάνω από την οποία η απόδοση του ταξινομητή μας θα μειώνεται (σχήμα 3.4).



Σχήμα 3.4: Η κατάρα της διαστατικότητας

Οι βασικές επιπτώσεις της κατάρα της διαστατικότητας μπορούν να διαχωριστούν ως εξής:

- Εκθετική αύξηση στον αριθμό των δειγμάτων που απαιτούνται για να διατηρηθεί η πυκνότητα των δειγμάτων
- Εκθετική αύξηση της πολυπλοκότητας της συνάρτησης προς υπολογισμό με αυξημένη διαστατικότητα
- Ενώ για μία διάσταση υπάρχουν πολλές διαθέσιμες συναρτήσεις, για συναρτήσεις πυκνότητας μεγάλων διαστάσεων μόνο η Gauss πολλών μεταβλητών είναι διαθέσιμη
- Ο άνθρωπος δυσκολεύεται να καταλάβει προβλήματα με περισσότερες από 3 διαστάσεις.

Σκεπτόμενοι λογικά, θα μπορούσαμε να υποθέσουμε ότι η αύξηση της διάστασης των χαρακτηριστικών δεν θα έπρεπε να μειώσει ποτέ την απόδοση, δεδομένου ότι παρέχεται ένα

μεγαλύτερο, ή τουλάχιστον το ίδιο, ποσό πληροφοριών-δεδομένων. Συνεπώς, το χειρότερο που θα μπορούσε να συμβεί, θα ήταν η απόδοση να παραμένει το ίδιο. Όπως όμως θα φανεί και στο πειραματικό κομμάτι, αυτό δυστυχώς δεν συμβαίνει. Η ακρίβεια του μοντέλου μπορεί να μειωθεί αν παρέχουμε περισσότερα δεδομένα στο σύστημα. Αυτή η συμπεριφορά οφείλεται στην πεπερασμένη ποσότητα των δεδομένων εκπαίδευσης που μπορεί να περιλαμβάνει με το μοντέλο. Θεωρητικά, θα μπορούσαμε να υποθέσουμε ότι τα δεδομένα εκπαίδευσης είναι άπειρα και έτσι, το μοντέλο καταλήγει να είναι άρτια εκπαιδευμένο και αποδοτικό κάτω από όλες τις συνθήκες. Στην πράξη αυτό δεν είναι δυνατόν, αν μάλιστα έχουμε επιλέξει ένα υπερβολικά πολύπλοκο μοντέλο, τότε είναι απίθανο το σύνολο των όλων των παραμέτρων μας να έχει εκτιμηθεί αποδοτικά. Από την άλλη μεριά, το μοντέλο δεν θα πρέπει να είναι πολύ απλοποιημένο, δεδομένου ότι αυτό θα εμπόδιζε το μοντέλο να προσεγγίσει την πολυπλοκότητα της ανθρώπινης ομιλίας.

Οι τεχνικές συμπίεσης, χρησιμοποιούν στατιστικές μεθόδους για τη μείωση της διάστασης των χαρακτηριστικών, μεγιστοποιώντας παράλληλα την πληροφορία που διατηρείται στο μειωμένο χώρο των χαρακτηριστικών. Μαθηματικά μπορούμε να εκφράσουμε αυτό εφαρμόζοντας ένα γραμμικό μετασχηματισμό,

$$y_i = B \cdot h_i$$

όπου το y_i συμβολίζει ένα χαρακτηριστικό διάνυσμα στο μειωμένο χώρο χαρακτηριστικών $y \in \Re_L$ και $h \in \Re_{2M}$ αντιπροσωπεύει ένα αρχικό χαρακτηριστικό διάνυσμα. Ο μετασχηματισμένος πίνακα B είναι ένας $2M \times L$ πίνακας, όπου L αποτελεί την νέα διάσταση για κάθε ιστορικό. Ο στόχος όλων των τεχνικών εξαγωγής χαρακτηριστικών είναι να βρεθεί το βέλτιστο B, βασιζόμενες σε κάποιο κριτήριο βελτιστοποίησης. Η Γραμμική Διαχωριστική Ανάλυση (Linear Discriminant Analysis) είναι μια τέτοια αποτελεσματική τεχνική.

3.3.2 Liner Discriminat Analysis

Η Γραμμική Διαχωριστική Ανάλυση ή Linear Discriminant Analysis ή LDA είναι μια τεχνική εξαγωγής χαρακτηριστικών που έχει εφαρμοστεί επιτυχώς σε πολλά στατιστικά προβλήματα αναγνώρισης. Σκοπός της είναι να χωρίσει δείγματα σε ομάδες μεγιστοποιώντας τη μεταξύ κλάσεων διαχωρισιμότητα (between class scatters) και την εντός κλάσης μεταβλητότητα (within class scatters), καθώς επίσης να μειώσει τις διαστάσεις ενώ θα διατηρήσεις όσο το δυνατόν πιο διακριτές τις κλάσεις.

Η τεχνική LDA βρίσκει τη βέλτιστο πίνακα μετασχηματισμού διατηρώντας τις περισσότερες από τις πληροφορίες που μπορούν να χρησιμοποιηθούν για τη διάκριση μεταξύ των

διαφόρων κατηγοριών. Συνεπώς, η ανάλυση απαιτεί τα δεδομένα να έχουν τις κατάλληλες ετικέτες της κατηγορίας, στην οποία ανήκουν (class labels). Για να επιτύχουμε αυτή τη διαδικασία, συνδέουμε τις κλάσεις των λέξεων με τις ετικέτες κατηγορίας και αναθέτουμε κάθε παρατηρούμενο διάνυσμα ιστορικών με την αντίστοιχη κατηγορία. Η LDA εκτιμά τις between και within class scatters, ώστε να βρεθεί ο βέλτιστος πίνακας μετασχηματισμού.

Για να διατυπώσουμε τη διαδικασία βελτιστοποίησης μαθηματικά, πρέπει να υπολογίσουμε το διάνυσμα των μέσων (mean vector) και το πίνακα συνδιακύμανσης (covarience matrix) για κάθε κατηγορία:

$$\overline{x_c} = \frac{1}{N_c} \sum_{i=1}^{N_c} x_i$$

$$W_c = \frac{1}{N_c} \sum_{i=1}^{N_c} (x_i - \overline{x_c})(x_i - \overline{x_c})^T$$

και για το πλήρες σύνολο δεδομένων (με όλες τις κατηγορίες που ομαδοποιήθηκαν μαζί)

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i$$

$$T = \frac{1}{N} \sum_{i=1}^{N} (x_i - \overline{x})(x_i - \overline{x})^T$$

Στους παραπάνω τύπους, το N δηλώνει το συνολικό αριθμό των training classses και N_c, αντιπροσωπεύει τον αριθμό των training classes της κατηγορίας c. Φυσικά υπάρχουν C κατηγορίες, όσες δηλαδή είναι και οι κλάσεις των λέξεων,

$$\sum_{c=1}^{C} N_c = N$$

Με αυτούς τους ορισμούς, μπορούμε να διατυπώσουμε εύκολα το κριτήριο βελτιστοποίησης (κριτήριο Fischer), δηλαδή

$$\widehat{B} = argmax_B \frac{|B^T \overline{T} B|}{|B^T W B|}$$

όπου

$$W = \frac{1}{N} \sum_{c=1}^{C} N_c W_c$$

Παρά το γεγονός ότι το κριτήριο αυτό μπορεί να φαίνεται περίπλοκο με την πρώτη ματιά, μπορεί εύκολα να γίνει κατανοητό. Ο αριθμητής αντιπροσωπεύει τη συνδιακύμανση των συγκεντρωμένων δεδομένων εκπαίδευσης στο μετασχηματισμένο χώρο χαρακτηριστικών.

Ο παρονομαστής αντιπροσωπεύει τη μέση συνδιακύμανση σε κάθε κατηγορία στο μετασχηματισμένο χώρο χαρακτηριστικών. Ως εκ τούτου, το κριτήριο στην πραγματικά προσπαθεί να μεγιστοποιήσει την «απόσταση» μεταξύ των κλάσεων, ελαχιστοποιώντας ταυτόχρονα το «μέγεθος» της κάθε μια από τις κλάσεις. Αυτό είναι ακριβώς αυτό που θέλουμε να επιτύχουμε, διότι το κριτήριο αυτό εγγυάται ότι θα διατηρηθεί το μεγαλύτερο μέρος της διακριτικής πληροφορίας στο μετασχηματισμένο χώρο χαρακτηριστικών.

Στην περίπτωσή μας, πρέπει να αντιμετωπίσουμε το πρόβλημα του μεγάλου μεγέθους των ιστορικών διανυσμάτων. Προκειμένου να εκτιμηθεί η προβολή του πίνακα B, εκτιμούμε τα στατιστικά στοιχεία που ο LDA χρησιμοποιεί για να υπολογίσει τους πίνακες διασποράς (scatter matrixes). Πρέπει να υπολογίσουμε την between-class scatter S_B και την within-class scatter S_w. Αρχικά, πρέπει να υπολογίσουμε δύο απαραίτητα στατιστικά στοιχεία για τον υπολογισμό των μετρικών, οι οποίες είναι οι εξής:

$$t_{1i} = \sum_{n \in i} x_n$$

$$t_{2i} = \sum_{n \in i} x_n x_n^T$$

όπου i είναι η κατηγορία της κλάσης των λέξεων δηλαδή $i = 1, \cdots, l$ και x_n είναι το διάνυσμα των ιστορικών.

Μετά την εκτίμηση αυτών των στατιστικών για όλα τα ιστορικά διανύσματα, υπολογίζουμε το mean vector για κάθε κλάση λέξεων κάθε κατηγορίας.

$$m_i = \frac{1}{n_i} \sum_{n \in i} x_n = \frac{1}{n_i} \cdot t_{1i}$$

όπου n_i είναι ο αριθμός από τα διανύσματα ιστορικών που ανήκουν στη κατηγορία i.

Στη συνέχεια, πρέπει να εκτιμήσουμε την between-scatter matrix, η οποία υπολογίζεται ως εξής:

$$\hat{S_B} = \sum_{i=c}^{l} n_i (m_i - m)(m_i - m)^T$$

όπου

$$m = \frac{1}{n} \sum_{i=1}^{l} n_i m_i$$

Για να εκτιμήσουμε την within-class scatter, θα πρέπει να υπολογίσουμε την S_i για κάθε κλάση κάθε κατηγορίας (D_i).

$$\begin{aligned} S_i &= \sum_{x \in D_i} (x - m_i)(x - m_i)^T \\ &= \sum_{x \in D_i} [xx^T - xm_i^T - m_i x + m_i m_i^T] \\ &= \sum_{x \in D_i} xx^T - \sum xm_i^T - m_i \sum x^T + n_i m_i m_i^T \\ &= \sum_{x \in D_i} xx^T - n_i m_i m_i^T \end{aligned}$$

έτσι μπορούμε να υπολογίσουμε,

$$\hat{S_w} = \sum_{i=1}^{l} S_i = \sum_{i=1}^{l} \sum_{x \in D_i} (x - m_i)(x - m_i)^T$$

Να σημειώνουμε ότι:

$$\hat{S_w} = B^T S_W B$$
$$\hat{S_b} = B^T S_B B$$

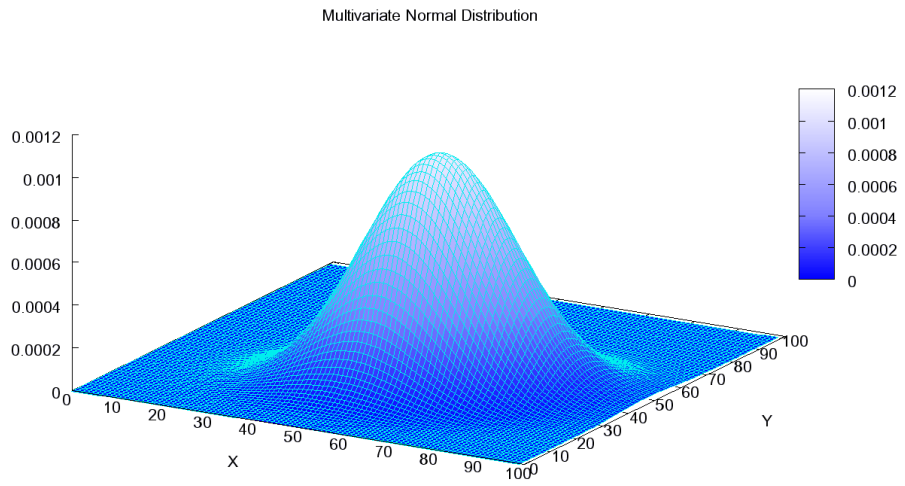
Καταλήγοντας, από το κριτήριο Fisher, ο πίνακα B υπολογίζεται,

$$\widehat{B} = argmax_B \frac{|B^T S_b B|}{|B^T S_w B|}$$

Το υπόλοιπο του μετασχηματισμού του πίνακα B γίνεται αντίστοιχα, έχοντας υπολογίζει όλα τα απαραίτητα μεγέθη. Με την εκτίμηση του projection matrix είμαστε σε θέση να προβάλουμε κάθε διάνυσμα ιστορικών στο νέο-διάστατο χώρο. Όπως και με τη τεχνική SVD, εκτελέσαμε διάφορα πειράματα για να καταλήξουμε στην μειωμένο χώρο χαρακτηριστικών $y \in \Re^L$, ώστε να προβάλλουμε κάθε διάνυσμα ιστορικών στη κατάλληλη διάσταση L.

3.4 Στατιστικές Μέθοδοι Συνεχών Γλωσσικών Μοντέλων

Για να οικοδομήσουμε ένα γλωσσικό μοντέλο στο συνεχή χώρο θα πρέπει να χρησιμοποιήσουμε κατανομές που ανήκουν στο συνεχή χρόνο. Η πιο διαδεδομένη και ευρύτερα χρησιμοποιημένη κατανομή για την εφαρμογή σε γλωσσικά μοντέλα, είναι η Gaussian κατανομή. Σε



Σχήμα 3.5: Multivariate Gaussian Distribution

πρώτη φάση, χρησιμοποιούμε μια πολυδιάστατη Gaussian κατανομή (Multivariate Gaussian Distribution) για την αρχική εκτίμηση των παραμέτρων και στη συνέχεια χρησιμοποιούμε μίξη Gaussian μοντέλων (Gaussian Mixture Models) και τον Expectation-Maximization αλγόριθμο. Τέλος, χρησιμοποιούμε παραμετρικά «δεμένες» τεχνικές για τη δημιουργία μίξης «δεμένων» Gaussian μοντέλων (Tied Gaussian Mixture Models).

3.4.1 Multivariate Gaussian Distribution

Στη θεωρία πιθανοτήτων και στατιστικής, η πολυδιάστατη κανονική κατανομή (multivariate normal distribution) ή multivatiate Gausian distribution (σχήμα 3.5), είναι μια γενίκευση της μονοδιάστατης (μονοπαραγοντικής) κανονικής κατανομής (normal distribution) σε υψηλότερες διαστάσεις. Ένας πιθανός ορισμός είναι, ότι, ένα τυχαίο διάνυσμα λέγεται ότι είναι p-κανονικά κατανεμημένων μεταβλητών (p-variate normally distributed), αν για κάθε γραμμικό συνδυασμό των στοιχείων p υπάρχει μια μεταβλητή κανονική κατανομής.

Ωστόσο, η σημασία αυτής της κατανομής, προέρχεται κυρίως από το πολυδιάστατο κεντρικό οριακό θεώρημα. Η multivariate normal distribution χρησιμοποιείται για να περιγράψει, τουλάχιστον κατά προσέγγιση, κάθε σύνολο (ενδεχόμενο) συσχετισμένων τυχαίων μεταβλητών πραγματικών τιμών, το οποίο συγκεντρώνονται (clustered) γύρω από τη μέση τιμή.

Μια multivatiate Gausian distribution ενός k-διάστατου τυχαίου διανύσματος $\mathbf{X} = [X_1, X_1, \cdots, X_k]$ μπορεί να γραφτεί με τον ακόλουθο συμβολισμό :

- $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

όπου $\boldsymbol{\mu}$, το k-διάστατο διάνυσμα της μέσης τιμής :

- $\boldsymbol{\mu} = [E[X_1], E[X_2], \cdots, E[X_k]]$

και $\boldsymbol{\Sigma}$, ο $k \times k$ πίνακας συνδιακύμανσης :

- $\boldsymbol{\Sigma} = [Cov[X_i, X_j]], i = 1, 2, \cdots, k; j = 1, 2, \cdots, k$

Εφόσον, μια multivariate normal distribution περιγράφει τις μεταβλητές που τείνουν να συγκεντρώνονται γύρω από τη μέση τιμή τους, με βάση το πολυδιάστατο κεντρικό οριακό θεώρημα, οποιαδήποτε τυχαία μεταβλητή μπορεί να περιγραφεί από την κανονική κατανομή, εάν έχει ένα μεγάλο σύνολο από παρατηρήσεις. Αυτός είναι και ο λόγος, που οι Gaussian κατανομές χρησιμοποιούνται συχνά στη στατιστική μοντελοποίηση και μοντελοποίηση της γλώσσας.

3.4.1.1 Multivariate Gaussian Language Model

Μετά τη συλλογή των mapped history data, χρησιμοποιούμε μια multivariate Gaussian distribution για κάθε κλάση λέξεων. Στη συνέχεια, υπολογίζουμε το διάνυσμα της μέσης τιμής και τον πίνακα συνδιακύμανσης κάθε κλάσης. Έτσι, το μοντέλο κάθε κλάσης εκπαιδεύεται με βάση τα ιστορικό του, y:

- $P(y|c) = \mathcal{N}(y; \mu_c, \Sigma_c)$ όπου μ_c και Σ_c είναι αποτελούν το διάνυσμα της μέσης τιμής και τον πίνακα συνδιακύμανσης.

Με την χρήση της παραπάνω κατανομής, εκτιμούμε την πιθανότητα κάθε ιστορικού δεδομένης της κλάσης στην οποία ανήκει. Όμως ο απώτερος σκοπός από τον υπολογισμό του παραπάνω μοντέλου, είναι η εκτίμηση της πιθανότητα κάθε κλάσης, δεδομένων των ιστορικών της. Χρησιμοποιώντας τον κανόνα του Bayes έχουμε:

$$P(c|y) = \frac{P(c)p(y|c)}{p(y)} = \frac{P(c)p(y|c)}{\sum_C^{c=1} P(c)p(y|c)}$$

όπου P(c) η πιθανότητα (unigram probability) της λέξης που ανήκει στην συγκεκριμένη κλάση.

Θα πρέπει να θεωρήσουμε, ότι ο όρος $P(c|y)$ πρέπει να αθροίζει στο ένα, $\forall c \in C$. Ένας κατάλληλος έλεγχος μπορεί να γίνει μετά την εκπαίδευση και πιο συγκεκριμένα για ένα σύνολο ιστορικών $Y = y_1 y_2 \cdots y_k$, ελέγχοντας εάν $\sum_{c \in C} P(c|y_n) = 1 \; \forall y_n \in H$

Οι παράμετροι του μοντέλου για την προσέγγιση αυτή, είναι η έξοδος της τεχνικής SVD που εφαρμόστηκε στο co-occurence matrix, ο πίνακας προβολής B του LDA καθώς και οι μέσες τιμές και οι πίνακες συνδιακύμανσής για κάθε κλάση.

3.4.2 Gaussina Mixure Models (GMM)

Τα GMMs είναι ίσως το πιο συχνά χρησιμοποιούμενο παράδειγμα από mixture distributions. Ένα Gaussian Mixture Model είναι μια παραμετρική συνάρτηση πυκνότητας πιθανότητας, που αναπαριστάται ως ένα σταθμισμένος γραμμικός συνδυασμός (μία μίξη) Gaussian κατανομών (components). Τα GMMs χρησιμοποιούνται συνήθως ως ένα παραμετρικό μοντέλο κατανομής πιθανότητας για μετρήσεις στο συνεχή χώρο ή σαν χαρακτηριστικά των ακουστικών και των γλωσσικών μοντέλων. Οι GMM παράμετροι υπολογίστηκαν από τα δεδομένα εκπαίδευσης, χρησιμοποιώντας τον επαναληπτικό Expectation-Maximization(EM) αλγόριθμο.

Ο EM, είναι ένας γενικός επαναληπτικός αλγόριθμος για εκπαίδευση μοντέλων με κρυφές μεταβλητές (όπως έχουμε και εδώ), που βελτιστοποιεί τη marginal likelihood των δεδομένων. Μεταπηδά μεταξύ δύο καταστάσεων, E και M, οι οποίες είναι υπεύθυνες για την εκτίμηση των κρυφών μεταβλητών δεδομένου του μοντέλου και στη συνέχεια, την εκτίμησης του μοντέλου δεδομένων αυτών των κρυφών μεταβλητών που υπολογίστηκαν αντίστοιχα.

Ένα Gaussian Mixture Model είναι ένα σταθμισμένο άθροισμα M συστατικών Gaussian πυκνοτήτων, όπως δίνεται και από την εξίσωση,

$$p(x|\lambda) = \sum_{k=1}^{K} c_k g(x|\mu_k, \Sigma_k)$$

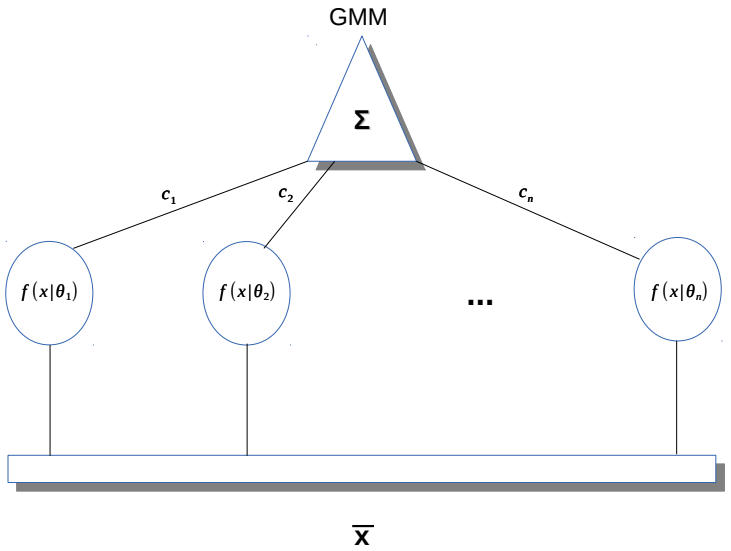
όπου x είναι ένα D-διαστατό διάνυσμα δεδομένων συνεχών τιμών (π.χ. μετρήσεις, χαρακτηριστικά), $c_k, k = 1, \cdots, K$, είναι τα βάρη των μειγμάτων (mixture weights) ή αλλιώς είναι η πιθανότητα ένα δείγμα να «έλκεται» από το k-οστό mixture component, δεδομένου ότι $\sum_{k=1}^{K} c_k = 1$ και $c_k > 0$ και τέλος, $g(x|\mu_k, \Sigma_k), k = 1, \cdots, K$, είναι τα component των Gaussian πυκνοτήτων. Κάθε component είναι μια D-μεταβλητών Gaussian συνάρτηση της μορφής,

$$g(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} exp\{-\frac{1}{2}(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)\}$$

όπου μ_i το διάνυσμα μέση τιμής και Σ_i τον πίνακα συνδιασποράς. Όπως ήδη δείξαμε, τα βάρη των μειγμάτων, ικανοποιούν τη συνθήκη ότι το άθροισμά τους είναι ένα. Το πλήρες Gaussian Mixure Model παραμετροποιείται από το διάνυσμα μέσης τιμής, το πίνακα συνδιακύμανσης και τα βάρη των μειγμάτων από όλα τα component πυκνοτήτων.
Αυτές οι παράμετροι συλλογικά, αναπαριστούνται από το συμβολισμό,

$$\lambda = \{c_k, \mu_k, \Sigma_k\} \text{ όπου } k = 1, \cdots, K$$

Οι πίνακες συνδιασποράς, Σ_i, μπορεί να είναι πλήρους βαθμού ή να περιορίζονται έτσι ώστε να είναι διαγώνιοι. Επιπλέον, οι παράμετροι μπορεί να έχουν μοιραστεί, ή να είναι δεμένοι, μεταξύ των Gaussian components, έχοντας ένα κοινό πίνακα συνδιασποράς για όλα τα components. Η επιλογή της διαμόρφωσης του μοντέλου (αριθμός από components, πλήρης στοιχεία ή διαγώνιοι πίνακες συνδιασποράς , και η παράμετρος «δεσίματος») συνήθως καθορίζεται από την ποσότητα των διαθέσιμων στοιχείων για την εκτίμηση των GMM παραμέτρων και το πώς τα GMM χρησιμοποιούνται σε μια εφαρμογή ενός γλωσσικού μοντέλου. Είναι επίσης σημαντικό να τονίσουμε, ότι επειδή τα Gaussian components ενεργούν από κοινού για να μοντελοποιηθεί η συνολική πυκνότητα πιθανότητας, οι πίνακες συνδιασποράς πλήρους βαθμού δεν είναι απαραίτητοι ακόμη και αν τα στοιχεία δεν είναι στατιστικός ανεξάρτητα. Ο γραμμικός συνδυασμός των διαγώνιων πινάκων συνδιακύμανσης, είναι ικανός στη μοντελοποίηση των συσχετίσεων μεταξύ των στοιχείων των χαρακτηριστικών διανυσμάτων. Η επίδραση της χρήσης ενός Gaussian σύνολου M πινάκων συνδιακύμανσης μπορούν να είναι εξίσου ισοδύναμοι, με τη χρήση ενός μεγαλύτερου συνόλου από Gaussians διαγώνιους πίνακες συνδιακύμανσης. Η τελική απόφαση για τη χρήση διαγώνιων ή ολόκληρων πινάκων συνδιασποράς, πραγματοποιήθηκε λαμβάνοντας υπόψη όλων των παραπάνω και εκτελώντας διάφορα πειράματα ώστε να ανιχνεύσουμε πότε εμφανίζονται τα καλύτερα αποτελέσματα. Τελικά, επιλέχτηκαν, οι διαγώνιοι πίνακες συνδιασποράς. Ένας άλλος λόγος επιλογής αυτών των πινάκων, ήταν ότι χρησιμοποιούνταν οι αντίστοιχοι πίνακες και στη διπλωματική η οποία αποτελεί το baseline μας, και έτσι αποσκοπούσαμε στο όσο το δυνατόν πιο συγκρίσιμα αποτελέσματα. Τα GMM χρησιμοποιούνται συχνά σε βιομετρικά συστήματα , κυρίως σε συνεχή συστήματα αναγνώρισης ομιλίας, λόγω της ικανότητάς τους ν΄ αναπαριστούν μια μεγάλη κλάση από δείγματα κατανομών.



Σχήμα 3.6: Gaussian Mixture Models

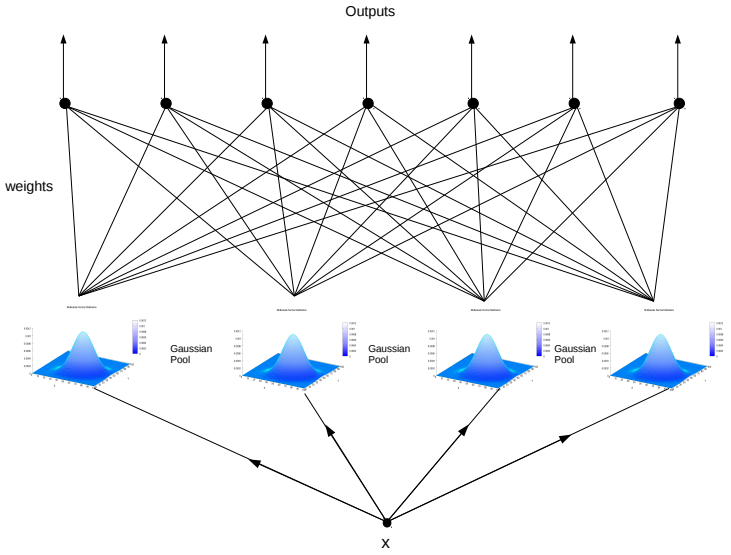
3.4.2.1 Gaussian Mixture Language Model(GMLM)

Όπως έχουμε ήδη αναφέρει, κάθε GMM αποτελείται από K_c κατανομές. Για να οικοδομήσουμε το μοντέλο μας, έχουμε εκπαιδεύσει ένα GMM για κάθε κλάση λέξεων χρησιμοποιώντας τον Expectation Maximization. Μετά την αντιστοίχιση των κλάσεων με τα ιστορικά τους, συλλέγουμε τα ιστορικά κάθε κλάσης και χτίζουμε τα μείγματα:

$$p(y|c) = \sum_{k=1}^{k_c} c_{c,k} \mathcal{N}(y, \mu_{c,k}, \Sigma_{c,k})$$

όπου K_c είναι ο αριθμός των components.

Χρησιμοποιούμε διαφορετικές τιμές για τον αριθμό K των μειγμάτων για κάθε εφαρμογή και εκπαιδεύουμε τις παραμέτρους του μοντέλου. Οι παράμετροι του μοντέλου για το GMLM, είναι η έξοδος του SVD που εφαρμόστηκε στο co-occurence matrix, ο πίνακας προβολής B του LDA καθώς και οι μέσες τιμές και οι πίνακες συνδιακύμανσής για κάθε κλάση. Μετά την εκτίμηση του μοντέλου, αξιολογούμε την λογαριθμική πιθανότητα και το perplexity των test data χρησιμοποιώντας τον κανόνα του Bayes, όπως αναλύσαμε παραπάνω.



Σχήμα 3.7: Gaussian Pool

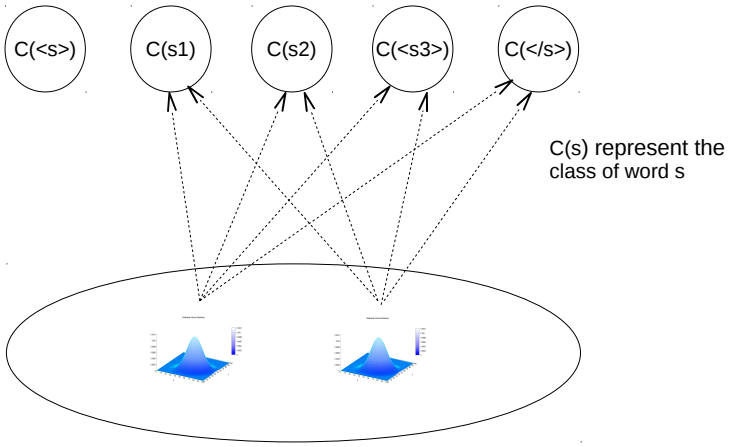
3.4.3 Tied Gaussian Mixture Models (T-GMM)

Τα GMMS χρησιμοποιούν διαφορετικό σύνολο κατανομών για κάθε μεταβλητή και κάθε ένα από αυτά, είναι παραμετροποιημένο από τα διανύσματα της μέσης τιμής, τους πίνακες συνδιακύμανσης και τα βάρη μιγμάτων από όλα τα componets των πυκνοτήτων. Είναι αναπόφευκτο ότι όσο η διάσταση και οι μεταβλητές αυξάνονται όλο και περισσότερο, τόσο και οι παράμετροι των GMMs αυξάνονται. Υπάρχει μια εξειδίκευση των GMMs όπου αντί να έχουμε χωριστά σύνολα Gaussian κατανομών για κάθε κλάση, έχουμε ένα κοινό σύνολο κατανομών το οποίο μπορεί να χρησιμοποιηθεί για όλες τις κλάσεις με διαφορετικά βάρη.

Ας υποθέσουμε ότι έχουμε ένα σύνολο κατανομών, μπορούμε να αναφερθούμε σε αυτήν ως μια Gaussian pool(3.7), η οποία είναι κοινή για όλες τις κλάσεις. Το Tied Gaussian mixture model είναι ένα σταθμισμένο άθροισμα από J components Gaussian densities, όπως φαίνεται και από την εξίσωση,

$$p(y|c) = \sum_{k=1}^{K} c_{w,j}\mathcal{N}(y, \mu_j, \Sigma_j)$$

Τα T-GMMs έχουν χρησιμοποιηθεί στην αναγνώριση προτύπων και σε εφαρμογές στατιστικής μοντελοποίησης, όπως η ακουστική μοντελοποίηση. Ένα από τα πλεονεκτήματά τους είναι ότι χρησιμοποιούν ένα μικρό σύνολο παραμέτρων για μεγάλο όγκο δεδομένων, και, κατά συνέπεια, η εκπαίδευση του μοντέλου είναι ακόμα πιο αποτελεσματική και λιγότερο πολύπλοκη.



Σχήμα 3.8: Parameter tying

3.4.3.1 Tied Mixture Language Model (TMLM)

Τα GMLMs προτείνονται ώστε να ξεπεραστούν τα προβλήματα που εμφανίζονται από τη χρήση των N-grams, όπως είναι η γενίκευση και προσαρμοστικότητα. Παρ΄ όλα αυτά, η μέθοδος αυτή έχει ένα μειονέκτημα όσον αφορά τον αριθμό των παραμέτρων, που είναι πολύ μεγάλος. Από την άλλη, τα TMLMs δεν παρουσιάζουν τέτοια προβλήματα στην εκτίμηση των παραμέτρων του μοντέλου, που έχουν τα GMLM. Το TMLM παρέχει μια μεγάλη ποικιλία παραμέτρων «δεμένες» πάνω στις κλάσεις (tying across classes), και ως εκ τούτου, επιτυγχάνει μια πολύ ισχυρή εκτίμηση των παραμέτρων. Επιπλέον, το TMLM μπορεί να εκτιμήσει την πιθανότητα οποιαδήποτε λέξης (της παρούσας κλάσης) η οποία έχει τουλάχιστον δύο εμφανίσεις στα δεδομένα εκπαίδευσης. Επίσης, ένας επιπλέον λόγος χρήσης του TMLM, είναι ότι στο GMM, τα δεδομένα εκπαίδευσης μπορεί να υποφέρουν από προβλήματα data-overfitting. Αυτό σημαίνει ότι ορισμένα δεδομένα ιστορικών, έχουν πολύ μικρές διακυμάνσεις, οι οποίες οδηγούν σε υπολογιστικά προβλήματα. Οι Tying parameters, όπως διανύσματα διασποράς (variance vectors) ή χρήση ολόκληρων μιγμάτων, χρησιμοποιούνται για να ξεπεραστεί το πρόβλημα αυτό.

Χρησιμοποιώντας tying τεχνικές για την εφαρμογή μας, δένουμε τα variance vectors κάθε κλάσης, έτσι ώστε να εκπαιδεύσουμε το μοντέλο μας. Έτσι, όλες οι κλάσεις χρησιμοποιούν τον ίδιο πίνακα συνδιασποράς Σ_j για κάθε κατανομή και διαφορετικά βάρη για κάθε μείγμα.

Οι παράμετροι του μοντέλου για το TMLM είναι η έξοδος του SVD που εφαρμόστηκε στο co-occurence matrix, ο πίνακας προβολής B του LDA καθώς και οι μέσες τιμές και οι

πίνακες συνδιακύμανσής για κάθε κλάση. Μετά την εκτίμηση του μοντέλου, αξιολογούμε την λογαριθμική πιθανότητα και το perplexity των test data χρησιμοποιώντας τον κανόνα του Bayes, όπως αναλύσαμε παραπάνω.

Κεφάλαιο 4

Ομαδοποίηση Λέξεων

Η ομαδοποίηση λέξεων είναι μια τεχνική που διαχωρίζει σύνολα λέξεων σε υποσύνολα από σημασιολογικά όμοιες (similar) λέξεις και αποτελεί μια όλο και πιο διαδεδομένη τεχνική που χρησιμοποιείται σε μια σειρά από natural language processing (NLP) εφαρμογές που λαμβάνουν υπόψη την έννοια της λέξης ή την δομική αποσαφήνισή της με σκοπό την ανάκτηση πληροφορίας και το φιλτράρισμα

4.1 Κατηγορίες Χαρακτηριστικών

Στη βιβλιογραφία της γλωσσολογίας (τόσο υπολογιστική όσο και μη-υπολογιστική), είναι ευρέως διαδεδομένο να ταξινομούμε τα χαρακτηριστικά των εγγράφων σε τρεις διαδεδομένες κατηγορίες χαρακτηριστικών, συντακτική (syntactic), σημασιολογική (semantic) και λεξιλογική (lexical) ομοιότητα (similarity). Είναι προφανές, ότι τα τμήματα της γλώσσας μπορούν να περιέχουν κι άλλες κατηγορίες χαρακτηριστικών, όπως για παράδειγμα στα μέσα μαζική ενημέρωσης οι λέξεις που εκφωνούνται μπορούν να έχουν αντίστοιχα φωνητικά χαρακτηριστικά, όπως τα αυτά που αναφέραμε παραπάνω, αλλά εμείς θα περιοριστούμε στις εκτιμήσεις που αφορούν τα χαρακτηριστικά των κειμένων.

Επομένως, καταλήγουμε σε τρεις βασικούς διαφορετικούς τύπους similarity οι οποίοι έχουν χρησιμοποιηθεί και μπορούν να αναλυθούν ως εξής:

- *syntactic similarity*: Αφορά τη συντακτική πτυχή της γλώσσας, που οριοθετεί το πως διαμορφώνουμε σωστές προτάσεις, τους επιφανειακούς κανόνες που διέπουν την οργάνωσή τους, και συγκεκριμένα, τη γραμματική και τη διαρθρωτική σχέση μεταξύ

των λέξεων (για παράδειγμα, πως ένα επίθετο χαρακτηρίζει συνήθως το ουσιαστικό που ακολουθεί μετά από αυτό)

- *semantic similarity*: Αφορά τις σημασιολογικές πτυχές της γλώσσας που είναι και το πιο πολύπλοκο, και καθορίζει τις προϋποθέσεις που διέπουν στο πως αντιλαμβανόμαστε και κατανοούμε το νόημα των προτάσεων και των λέξεων

- *lexical similarity* [1]: Αφορά τις λεξικές πηγές της γλώσσας, που διέπουν την ομαδοποίηση των λέξεων από το θέμα, και συνήθως αφορά κάποια έννοια του hypernym (something more general)/hyponym (something more specific)[2].

Και οι τρεις τύποι ομοιότητας, υπολογίζονται με διαφορετικές μεθόδους και χρησιμοποιούνται σε ένα ευρύ φάσμα εφαρμογών NLP. Στη δική μας δουλειά, έχει δοθεί βάρος σε lexical similarity. Πιο πιο συγκεκριμένα, η ομαδοποίηση στηρίζεται στις εμφανίσεις, των λέξεων και τον bigram 2.1 που δημιουργούνται μεταξύ των λέξεων του corpus που περιγράφουμε στην παρακάτω ενότητα.

4.2 Αντιστοίχιση λέξεων σε διανύσματα

Το συντριπτικό ποσοστό των αλγορίθμων ομαδοποίησης, χρησιμοποιούν εργαλεία διαφορικού λογισμού ή άλλα μαθηματικά εργαλεία με αποτέλεσμα την ανάγκη αντιστοίχισης κάθε λέξης σε ένα διάνυσμα (word mapping). Για την υλοποίηση αυτή της διαδικασίας, αρχικά, υπολογίζουμε τον πίνακα word co-occurence αντίστοιχα με το τρόπο που περιγράψαμε αναλυτικά στο κεφάλαιο 3.2.2. Πιο συγκεκριμένα, σε αυτήν την περίπτωση, υπολογίζουμε τον πίνακα συν-εμφανίσεων λέξεων (word co-occurences) e_{ij} που απεικονίζει τη σχέση μεταξύ των λέξεων υπολογίζοντας το bigram για κάθε λέξη i kai j. Έτσι, κάθε στοιχείο (element) e_{ij} είναι ο αριθμός των φορών που η λέξη i ακολουθείται από τη λέξη j, στα δεδομένα εκπαίδευσης.

[1] Ορισμένοι συγγραφείς, όπως ο (Allen 1987, PP6-8) συμπεριλαμβάνουν σε αυτή, ένα ακόμα σύνολο από κατηγορίες χαρακτηριστικών, την κατηγορία pragmatic features (πραγματικά χαρακτηριστικά), που αφορά το τρόπο με τον οποίο χρησιμοποιούνται οι λέξεις και οι εκφράσεις που περιέχονται σε ένα έγγραφο. Αυτά, χωρίς αμφιβολία, είναι σημαντικά χαρακτηριστικά για τον προσδιορισμό του νοήματος του κειμένου, γι΄ αυτό το λόγο ακριβώς θεωρούμε τα χαρακτηριστικά αυτά ότι είναι semantic (σημασιολογικά)

[2] Η λέξη αιλουροειδές είναι Hypernym τόσο του λύγκα όσο και της λέξης γάτα, τα οποία είναι hyponyms της αρχικής λέξης, επειδή κατέχει μια σχέση ISA (δηλαδή κυριολεκτώντας, "is a") με αυτές, ως προς τη σχέση ότι μια γάτα είναι ένα αιλουροειδές, εξίσου λύγκα είναι ένα αιλουροειδές, αλλά δεν έχει νόημα να πούμε ότι ένα αιλουροειδές είναι μια γάτα ή ένα λύγκα

Κάθε γραμμή αναπαριστά το διάνυσμα κάθε λέξης (word vector) και οι στήλες αντιπροσωπεύουν τα ιστορικά. Για την αντιμετώπιση των μηδενικών στοιχείων, γίνεται smoothing σύμφωνα με τη σχέση $e_{ij} = log(1 + e_{ij})$. Όσον αφορά το μεγάλο αριθμό των στοιχείων, χρησιμοποιούμε διάφορες τεχνικές μείωσης διαστάσεων. Αναλυτικότερα, εφαρμόζουμε στον word co-oocurence matrix μια τεχνική συμπίεσης δεδομένων, η οποία ονομάζεται Ανάλυση στη Βάση των Ιδιάζουσων Τιμών (SVD) την οποία έχουμε ήδη περιγράψει σε προηγούμενα κεφάλαια. Η κατανόηση της δημιουργία του word co-oocurence matrix όσο και της εφαρμογής της τεχνική SVD έχουν αναλυθεί πλήρως στο κεφάλαιο 3.

Στη περίπτωση χρήση του μικρού λεξιλογίου $V = 2700$ λέξεις, κάθε λέξη περιγράφεται από ένα διάνυσμα $M = 100$ διαστάσεων μετά την εφαρμογή του SVD. Για την περίπτωση του μεγάλου λεξιλογίου, με χρήση του συνόλου των διαφορετικών λέξεων (V = 58831) που εμφανίζονται στα δεδομένα, κάθε λέξη περιγράφεται από ένα διάνυσμα M = 300 διαστάσεων μετά την εφαρμογή του SVD. Θα μπορούσαμε να χρησιμοποιήσουμε μόνο τον word co-oocurence matrix για να αντιστοιχίσουμε σε κάθε λέξη ένα διάνυσμα, ώστε στη συνέχεια να ομαδοποιήσουμε τις λέξεις. Η εφαρμογή της τεχνική συμπίεσης δεδομένων SVD αποσκοπούσε τόσο στην μείωση της αρχικής διάστασης των διανυσμάτων με απώτερο σκοπό την αντιμετώπιση της μεγάλη πολυπλοκότητα λόγο των πολλών διαστάσεων, όσο και στην εξομάλυνση των τιμών κάθε διανύσματος (υπερβολικά μεγάλες τιμές, μηδενικά) απαλείφοντας συγχρόνως τα πλεονάζοντα χαρακτηριστικά (features).

4.3 Αλγόριθμοι Ομαδοποίηση Λέξεων

Οι αλγόριθμοι ομαδοποίησης μπορούν να εκληφθούν ως σχήματα που προσδιορίζουν λογικές ομαδοποιήσεις, εξετάζοντας μόνο ένα μικρό κλάσμα του συνόλου των δυνατών διαμερίσεων. Το αποτέλεσμα εξαρτάται από τον συγκεκριμένο αλγόριθμο και από τα κριτήρια που υιοθετήθηκαν. Έτσι, ένας αλγόριθμος ομαδοποίησης είναι μια διαδικασία εκμάθησης, η οποία προσπαθεί να προσδιορίσει τα συγκεκριμένα χαρακτηριστικά των ομάδων, που σχηματίζουν τα διανύσματα συνόλου δεδομένων. Οι αλγόριθμοι ομαδοποίησης μπορούν να χωριστούν σε πληθώρα κατηγοριών. Οι κατηγορίες αλγορίθμων που χρησιμοποιήσαμε εμείς είναι οι ακόλουθες:

Ιεραρχικοί αλγόριθμοι ομαδοποίησης (Hierarchical clustering algorithms). Οι αλγόριθμοι αυτοί χωρίζονται περαιτέρω σε:

- *Συσσωρευτικούς αλγορίθμους (agglomerative algorithms).* Οι αλγόριθμοι αυτοί πα-

ράγουν μια ακολουθία ομαδοποιήσεων όπου ο αριθμός των ομάδων, μειώνεται σε κάθε βήμα. Η ομαδοποίηση που παράγεται σε κάθε βήμα προκύπτει από την προηγούμενη συγχωνεύοντας δυο ομάδες σε μια. Οι αντιπροσωπευτικότεροι συσσωρευτικοί αλγόριθμοι είναι οι αλγόριθμοι *απλού δεσμού (single link)* και *πλήρους δεσμού (complete link)*. Οι συσσωρευτικοί αλγόριθμοι είναι δυνατόν να διαιρεθούν περαιτέρω στις ακόλουθες υποκατηγορίες.

- *Αλγόριθμοι που προκύπτουν από τη θεωρία πινάκων*. Παραδείγματα τέτοιων αλγορίθμων που θα αναλύσουμε παρακάτω, αποτελούν οι GAAG 4.5.1 και Bagglo 4.6.4.
- *Αλγόριθμοι που προκύπτουν από τη θεωρία γράφων*. Παράδειγμα τέτοιου αλγορίθμου που θα αναλύσουμε παρακάτω, αποτελεί ο Graph/KNN 4.6.5.

Οι αλγόριθμοι αυτοί είναι κατάλληλοι για την ανεύρεση τόσο επιμήκων ομάδων (όπως συμβαίνει στην περίπτωση του αλγορίθμου απλού δεσμού) όσο και συμπαγών ομάδων (όπως συμβαίνει στη περίπτωση του αλγορίθμου πλήρους δεσμού)

- *Διαιρετικούς αλγορίθμους (divisive algorithms)*. Οι αλγόριθμοι αυτοί ακολουθούν την αντίθετη κατεύθυνση, δηλαδή παράγουν μια ακολουθία ομαδοποιήσεων, όπου ο αριθμός τον ομάδων, αυξάνει σε κάθε βήμα. Η ομαδοποίηση που παράγεται, σε κάθε βήμα, προκύπτει από την προηγούμενη χωρίζοντας μια ομάδα σε δυο μικρότερες. Παραδείγματα τέτοιων αλγορίθμων που θα αναλύσουμε παρακάτω αποτελούν οι Rbr 4.6.2 και Direct 4.6.3.

Αλγόριθμοι ομαδοποίησης που βασίζονται στη βελτιστοποίηση συνάρτησης κόστους (cost function optimiazation clustering algorithm). Αυτή η κατηγορία περιέχει αλγορίθμους όπου ο όρος «λογική ομαδοποίηση» ποσοτικοποιείται με τη χρήση μιας συνάρτηση κόστους, J, με βάση την οποία αξιολογούνται οι ομαδοποιήσεις. Συνήθως, ο αριθμός των ομάδων θεωρείται σταθερός. Οι περισσότεροι από αυτούς τους αλγορίθμους χρησιμοποιούν εργαλεία διαφορικού λογισμού και παράγουν διαδοχικές ομαδοποιήσεις στην προσπάθεια τους να βελτιστοποιήσουν την J. Τερματίζουν όταν φτάσουν σ΄ ένα τοπικό βέλτιστο της J. Οι αλγόριθμοι αυτής της κατηγορίας καλούνται επίσης και *επαναληπτικά σχήματα βελτιστοποίησης συνάρτησης κόστους (iterative cost function optimization schemes)*. Η κατηγορία αυτή περιλαμβάνει τις ακόλουθες υποκατηγορίες:

- *Αυστηροί αλγόριθμοι ομαδοποίησης (hard/crisp clustering algorithms)*, όπου ένα διάνυσμα ανήκει αποκλειστικά σε μια συγκεκριμένη ομάδα. Η καταχώρηση διανυσμάτων στις ομάδες γίνεται με έναν βέλτιστο τρόπο, σύμφωνα με το χρησιμοποιούμενο κριτήριο βελτιστοποίησης. Ο πιο δημοφιλής αλγόριθμος της κατηγορίας αυτής, που θα αναλύσουμε παρακάτω, είναι ο λεγόμενος αλγόριθμος Isodata ή αλγόριθμος K-means 4.5.1.

- *Πιθανοτικοί αλγόριθμοι ομαδοποίησης (probabilistic clustering algorithms)*, οι οποίοι είναι ένα είδος αυστηρών αλγορίθμων που χρησιμοποιούν στοιχεία από τη θεωρία ταξινόμησης κατά Bayes. Έτσι, κάθε διάνυσμα, x, καταχωρείται στην ομάδα C_i για την οποία η εκ των υστέρων (a-posteriori) πιθανότητα $P(C_i|x)$ είναι η μεγαλύτερη. Οι πιθανότητες αυτές εκτιμώνται μέσω μιας κατάλληλα ορισμένης διαδικασίας βελτιστοποίησης.

- *Αλγόριθμοι ομαδοποίησης ασαφούς λογικής (fuzzy clustering algorithms)* όπου ένα διάνυσμα ανήκει σε μια συγκεκριμένη ομάδα μέχρι ενός ορισμένου βαθμού.

- *Αλγόριθμοι ομαδοποίησης στη βάση των ενδεχομένων (possibilistic clustering algorithms)*. Εδώ μετρούμε το ενδεχόμενο ένα διάνυσμα x να ανήκει στην ομάδα C_i.

- *Αλγόριθμοι ανίχνευσης ορίων (boundary detection algorithms)*. Αντί να προσδιορίζουν τις ομάδες μέσω των διανυσμάτων χαρακτηριστικών, οι αλγόριθμοι αυτοί προσαρμόζουν επαναληπτικά τα όρια των περιοχών όπου βρίσκονται οι ομάδες. Αυτοί οι αλγόριθμοι, παρότι αναπτύσσονται στα πλαίσια μιας φιλοσοφίας βελτιστοποίησης συνάρτησης κόστους, είναι διαφορετικοί από αυτούς που αναφέρθηκαν προηγουμένως. Όλα τα προηγούμενα σχήματα χρησιμοποιούσαν αντιπροσώπους ομάδων και ο στόχος είναι να τους τοποθετήσουν στο χώρο σύμφωνα με ένα βέλτιστο τρόπο. Αντίθετα, οι αλγόριθμοι ανίχνευσης ορίων αναζητούν τρόπους βέλτιστης τοποθέτησης ορίων μεταξύ των ομάδων.

4.4 Ομαδοποίηση με χρήση αλγορίθμου Brown-SRILM

Το εργαλείο SRI Lnaguage Model (SRILM) είναι μια συλλογή από C++ βιβλιοθήκες, εκτελέσιμα προγράμματα, καθώς και βοηθητικών scripts που έχουν σχεδιαστεί για να επιτρέπουν τόσο την παραγωγή όσο τον πειραματισμό με στατιστικά γλωσσικά μοντέλα για την αναγνώριση φωνής και άλλες εφαρμογές. Αυτό το toolkit υποστηρίζει τη δημιουργία και την

αξιολόγηση διαφορετικών τύπων γλωσσικά μοντέλα που βασίζονται σε N-gram στατιστικά στοιχεία, καθώς και άλλες σχετικές λειτουργίες, όπως η εξομάλυνση και η δημιουργία μοντέλων που βασίζονται σε κλάσεις, το statistical tagging και ο χειρισμός των N-καλύτερων λιστών και πλεγμάτων (lattices) από λέξεις.

Στο συγκεκριμένο σημείο, η χρήση του εργαλείου θα περιοριστεί στην δημιουργία μοντέλων βασισμένων σε κλάσεις. Πιο συγκεκριμένα, θα εκμεταλλευτούμε το κομμάτι του μοντέλου που αφορά την ομαδοποίηση. Τα μοντέλα αυτά, που αποτελούν N-grams πάνω σε κλάσεις λέξεων (word classes), είναι ένας αποτελεσματικός τρόπος για την αύξηση της ανθεκτικότητας (robustness) των γλωσσικών μοντέλων και της ενσωμάτωσης γνώσης, π.χ. με τον καθορισμό κλάσεων από λέξεις που αντικατοπτρίζουν τη σημασιολογία του task. Επίσης, το SRILM επιτρέπει στα μέλη των κλάσεων να είναι multiword (πολυ-λεκτικά) strings (π.χ. το «Σαν Φρανσίσκο» μπορεί να είναι ένα μέλος της κλάσης «Πόλη-Όνομα»). Αυτό, σε συνδυασμό με το γεγονός ότι οι λέξεις μπορούν να ανήκουν σε περισσότερες από μία κλάσεις, απαιτεί τη χρήση δυναμικού προγραμματισμού για να την αξιολόγηση ενός class N-gram.

Ειδικότερα στην δική μας υλοποίηση, οι κλάσεις λέξεων ορίζονται από την εντολή-εργαλείο "ngram-class", η οποία οδηγεί στην εξαγωγή στατιστικών bigram χρησιμοποιώντας τον αλγόριθμο ομαδοποίησης του Brown (αλγόριθμος 1, δημοσίευση [11]).

Σύμφωνα από έρευνα που έγινε σχετικά με το SRILM, δεν γνωρίζουμε καμιά διαθέσιμη υλοποίηση για ομαδοποιήσεις λέξεων βασισμένες σε trigram μοντέλα. Στηριζόμενοι στην εργασία [12], παρουσιάζεται ότι η βελτίωση της ακρίβειας του μοντέλου με χρήση trigram, σε σχέση με τα bigram, που επιφέρει στις κλάσεις, είναι πολύ μικρή.

Στο πρακτικό κομμάτι, χρησιμοποιήσαμε το ngram-class tool του SRILM για να ομαδοποιήσουμε τις λέξεις μας σε κλάσεις, το οποίο βασίζεται πάνω σε bigram μοντέλα. Ακολουθεί ένα παράδειγμα χρήσης:

$$\begin{aligned} ngram - class & - vocab\ Lexicon.file \\ & - text\ train.txt \\ & - numclasses\ C \\ & - classes\ CLM\_C.classes \end{aligned}$$

Παράμεροι:
-vocab: είναι η λίστα όλων των λέξεων του λεξιλογίου
-text: είναι το αρχείο κειμένου που περιέχει τα δεδομένα εισόδου, με χρήση δείκτη-χαρακτήρα

Αλγόριθμος 1 Brown Clustering Algorithm

Input: $m, corpus$

Output: $c_1, c_2, \cdots, c_m$

1: Αρχή
2: εύρεση: μεγέθους λεξιλογίου V, class co-occurence matrix $P(C, C')$, unigrams $P(w)$, class probability $P(C)$
3: συγκεντρώνουμε τις m συχνότερες λέξεις και τοποθετούμε κάθε μια στις $c_1, c_2, \cdots, c_m$ κλάσεις
4: **for** $i = m + 1...|V|$ **do**
5: δημιουργία μιας καινούργιας κλάσης, c_{m+i} για την i_{th} συχνότερη λέξη (έχουμε $m + 1$ κλάσεις).
6: Επιλογή 2 κλάσεων από τις $c_1, c_2, \cdots, c_{m+1}$ για να συγχωνευθούν
7: Επιλογή της συγχώνευσης που δίνει την μέγιστη τιμή για το κριτήριο συγχώνευσης $Quality(C) = \sum_{C,C'} P(C, C') log \frac{P(C,C')}{P(C)P(C')} + \sum_w P(w) log P(w)$. Έχουμε πάλι m κλάσεις
8: **end for**
9: **return** $c_1, c_2, \cdots, c_m$
10: Τέλος

για την αρχή της πρότασης και έναν για το τέλος της πρότασης (<s> και </s> αντίστοιχα)
-numclasses: θέτουμε τον αριθμό των κλάσεων
-classes: το αρχείο εξόδου περιέχει τις κλάσεις (μαζί με τις λέξεις που περιέχει κάθε κλάση) οι οποίες δημιουργήθηκαν, καθώς και την πιθανότητα εμφάνισης κάθε λέξης στη συγκεκριμένη κλάση

-Έξοδος αρχείου CLM_C.classes

CLASS-00001 0.998834 <unk>
CLASS-00003 0.197272 martin
CLASS-00003 0.13064 philip
CLASS-00003 0.0823715 northwest
CLASS-00003 0.106506 maclean
CLASS-00003 0.483211 every

CLASS-00033 0.065878 wrote
CLASS-00033 0.125493 adds
CLASS-00033 0.0605428 indicated
CLASS-00033 0.250058 added
CLASS-00033 0.0338669 argues
CLASS-00033 0.0951055 noted
CLASS-00033 0.0415217 warned
CLASS-00033 0.0730689 believes
CLASS-00033 0.046161 argued
CLASS-00033 0.0317792 contends
CLASS-00033 0.0498724 thinks
CLASS-00033 0.0401299 predicted
CLASS-00033 0.0299235 predicts
CLASS-00033 0.0565994 suggests
⋮
CLASS-00135 0.95406 time
CLASS-00135 0.0459399 stage
CLASS-00139 0.37779 above
CLASS-00139 0.333705 near
CLASS-00139 0.288504 below
CLASS-00140 0.0221801 aside
CLASS-00140 0.97782 up
CLASS-00145 0.82725 he
CLASS-00145 0.17275 she
⋮

Μετά την επεξεργασία του αρχείου εξόδου από το εργαλείο SRILM, για εμφανέστερη απεικόνιση, παραθέτουμε κομμάτια των κλάσεων που δημιουργήθηκαν, ώστε να καταστήσουμε εμφανές την ύπαρξη σημασιολογικής, λεξιλογικής ή συντακτικής σχέσης μεταξύ των λέξεων και επομένως την ορθή ομαδοποίηση των λέξεων (Τάση Ομαδοποίησης-Clustering Tendency). Είναι προφανές, ότι δεν μπορούνε να παρουσιάσουμε την πλήρης ομαδοποίηση λόγο του μεγάλου πλήθους των δεδομένων.

Οι χαρακτήρες που αποτελούν δείκτες για την αρχή και το τέλος της πρότασης (<s> και </s>), καθώς και ο χαρακτήρας <unk> λαμβάνονται υπόψη σαν ξεχωριστές, διακριτές κλάσεις, περιορισμός όπου εφαρμόστηκε σε όλες τις τεχνικές ομαδοποίησης που υλοποιή-

θηκαν.

-Έξοδος αρχείου CLM_C.classes μετά από επεξεργασία για καλύτερη απεικόνιση (η τελευταία δεξιά στήλη αντιστοιχεί στον κωδικό-όνομα κάθε κλάσης):

martin, philip, northwest, maclean, every 2
wrote, adds, indicated, added, argues, noted, warned, believes, argued, contends, thinks, predicted, predicts, suggests 3
time, stage 19
above, near, below 20
aside, up 21
he, she 22
saturday, wednesday, thursday, tuesday, monday, yesterday, sunday, friday 363
⋮
<unk> 513
</s> 514
<s> 515

4.5 Ομαδοποίηση με χρήση του εργαλείου NLTK

4.5.1 Τι είναι το NLTK

Το εργαλείο Natural Language Toolkit είναι μια κορυφαία πλατφόρμα για την οικοδόμηση Python προγραμμάτων που χρησιμοποιούνται σε εργασίες σχετικά με δεδομένα ανθρώπινης γλώσσας. Παρέχει εύκολη χρήση σε διεπαφές για πάνω από 50 σώματα (corpora) και πόρους λεξιλογίων, όπως το WordNet, μαζί με μια σειρά από βιβλιοθήκες επεξεργασίας κειμένου για ταξινόμηση (classification), tokenization, stemming, tagging, parsing και σημασιολογικό συλλογισμό (semantic reasoning).

Χάρη σε ένα μικρό οδηγό, εισάγει βασικά στοιχεία προγραμματισμού, παράλληλα με θέματα που αφορούν την υπολογιστική γλωσσολογία, το NLTK είναι κατάλληλο για γλωσσολόγους, μηχανικούς, φοιτητές , εκπαιδευτικούς, ερευνητές και τους χρήστες του κλάδου.

Ο λόγος που χρησιμοποιήθηκε αυτό το εργαλείο, είναι για να επωφεληθούμε από τις υλοποιήσεις που παρέχει σε αλγορίθμους ομαδοποίησης. Πιο συγκεκριμένα, μέσω αυτής της πλατφόρμας, υλοποιήσαμε τους αλγορίθμους ομαδοποίησης k-means και Group Average

Agglomerative Clustering(GAAC) που αναλύουμε στα επόμενα δυο κεφάλαια.

4.5.2 K-means Αλγόριθμος

Πρόκειται για έναν από τους πιο γνωστούς αλγορίθμους ομαδοποίησης με μεγάλο αριθμό βελτιστοποιήσεων και διατυπώσεων. Η δική μας υλοποίηση όπως λογικά προκύπτει, βασίζεται στην υλοποίηση του NLTK [13]. Αναλυτικότερα, ο αλγόριθμος ομαδοποίησης K-means ξεκινά με K αυθαίρετα επιλεγμένες μέσες τιμές όπου στη συνέχεια κατανέμει κάθε διάνυσμα λέξης στη κλάση με την πλησιέστερη μέση τιμή (χρήση Ευκλείδειας απόστασης). Στη συνέχεια, υπολογίζει εκ νέου τις μέσες τιμές κάθε κλάσης ως το κέντρο βάρους των διανυσμάτων κάθε κλάσης. Αυτή η διαδικασία επαναλαμβάνεται έως ότου τα μέλη της κλάσης σταθεροποιηθούν. Αποτελεί έναν αλγόριθμο αναρρίχησης λόφων που μπορεί να συγκλίνει σε ένα τοπικό μέγιστο. Επομένως, η ομαδοποίηση συχνά επαναλαμβάνεται με τυχαίες αρχικές μέσες τιμές και σε κάθε επόμενη ομαδοποίηση επιλέγονται οι πιο συχνά εμφανιζόμενες μέσες τιμές που έχουν προκύψει από την προηγούμενη εκτέλεση. Στην δικιά μας υλοποίηση, επιλέξαμε η εκτέλεση του αλγορίθμου να πραγματοποιηθεί 20 φορές ώστε να εξασφαλίζουμε μεγαλύτερη ακρίβεια στις τυχαίες αρχικές μέσες τιμές. Έτσι, επιλέγεται η αποδοτικότερη από τις 20 ομαδοποιήσεις, δηλαδή αυτή με το μεγαλύτερο τοπικό μέγιστο.

Όπως συμβαίνει με όλους τους αλγορίθμους που χρησιμοποιούν σημειακούς αντιπροσώπους, ο αλγόριθμος K-means είναι κατάλληλος για την ανάδειξη συμπαγών ομάδων. Μια ακολουθιακή έκδοση του αλγορίθμου, προκύπτει αν η ενημέρωση των αντιπροσώπων λαμβάνει χώρα αμέσως μετά τον προσδιορισμό του αντιπροσώπου που βρίσκεται εγγύτερα στο τρέχον διάνυσμα x_i. Είναι σαφές ότι το αποτέλεσμα αυτής της έκδοσης του αλγορίθμου εξαρτάται από την διάταξη με την οποία τα διανύσματα εξετάζονται από τον αλγόριθμο.

Ένα σημαντικό πλεονέκτημα του αλγορίθμου k-means είναι η υπολογιστική του απλότητα, η οποία τον καθιστά ελκυστικό υποψήφιο για ποικιλία εφαρμογών. Αυτή η υπολογιστική απλότητά του αποτέλεσε πηγή έμπνευσης πολλών συγγραφέων, οι οποίοι πρότειναν ένα σημαντικό αριθμό παραλλαγών του προκειμένου να αντιμετωπίσουν τα μειονεκτήματα του.

Αλγόριθμος 2 K-means Clustering Algorithm

Input: $n, c, \mu_1, \mu_2, \cdots, \mu_c$

Output: $\mu_1, \mu_2, \cdots, \mu_c$

Αρχή
αρχικοποίηση των n, c, $\mu_1, \mu_2, \cdots, \mu_c$ (τυχαία επιλεγμένα)
repeat
  ταξινόμηση των n σημείων με βάση τα κοντινότερα μ_i
  επανάληψη υπολογισμού μ_i
until καμιά αλλαγή στις τιμές των μ_i
return $\mu_1, \mu_2, \cdots, \mu_c$
Τέλος

Clustering Tendency

-Έξοδος αρχείου ομαδοποίησης του αλγορίθμου k-means (η δεξιά στήλη αντιστοιχεί στον κωδικό-όνομα κάθε κλάσης):

those, anyone, someone, you 4
everybody, everyone, weve, ive, id, theyve 10
publishing, founder, owner 12
yesterday, friday, monday, thursday, tuesday, wednesday 85
pursue, expand, join, eliminate, grow, maintain, build, create, develop, replace, determine, attract, improve, impose, deliver, earn, save, file 226
sachs, noting, citing, reflecting 227
⋮
<unk> 511
</s> 512
<s> 513

4.5.3 Group Average Agglomerative Clustering (GAAC) Αλγόριθμος

Όπως έχουμε αναφέρει και παραπάνω, ο συσσωρευτικός αλγόριθμος ομαδοποίησης μέσης τιμής (GAAC) ανήκει στην κατηγορία των ιεραρχικών αλγόριθμων ομαδοποίησης, και πιο συγκεκριμένα στους συσσωρευτικούς αλγορίθμους. Ο αλγόριθμος GAAC που υλο-

ποιείται από το NLTK [14] ξεκινά θεωρώντας κάθε ένα από τα N διανύσματα λέξεων σαν μια μοναδική κλάση. Σε κάθε βήμα εκτέλεσης, συγχωνεύει επαναληπτικά, ζευγάρια από κλάσεις που έχουν το πιο κοντινό κέντρο βάρους (closest centroids). Αυτό συνεχίζεται έως ότου υπάρχει μόνο μία κλάση. Η σειρά αυτών τον συγχωνεύσεων, δημιουργεί ένα δενδρόγραμμα-dendogram (Pattern Recognition Book [4] Chapter 12.2.1): ένα δέντρο με τις προγενέστερες συγχωνεύσεις να βρίσκονται «χαμηλότερα» από τις μεταγενέστερες συγχωνεύσεις. Η σύνθεση ενός δεδομένου αριθμού κλάσεων c, όπου $1 \leq c \leq N$, με N τον αριθμό των διανυσμάτων των λέξεων, μπορεί να βρεθεί με την κοπή του dendogram σε βάθος c.

Ο συνολικός αριθμός πράξεων που απαιτούνται από ένα agglomerative αλγοριθμικό σχήμα είναι ανάλογος του N^3. Ωστόσο, η ακριβής πολυπλοκότητα του αλγορίθμου εξαρτάται και από τον ορισμό της μετρικής που καθορίζει την μέθοδο συγχώνευσης μεταξύ δυο κλάσεων. Στη δική μας περίπτωση, η πολυπλοκότητα ενός group average (average of all similarities) περιορίζεται σε $N^2 logN$. Αναλυτικότερα, αυτός ο αλγόριθμος ομαδοποίησης, χρησιμοποιεί τη μετρική cosine similarity, η οποία επιτρέπει την αποτελεσματική επιτάχυνση της διαδικασίας ομαδοποίησης, και θεωρείται η καλύτερη επιλογή συσσωρευτικού αλγορίθμου για τις περισσότερες εφαρμογές.

Αλγόριθμος 3 GAAC Clustering Algorithm

Input: $c, h_1, h_2, \cdots, h_c$

Output: $C_1, C_2, \cdots, C_c$

1: Αρχή
2: θέτουμε κάθε διάνυσμα λέξης να αποτελεί μια κλάση
3: Υπολογισμός της μέσης «απόστασης» για κάθε κλάση C_i και C_j
4: **repeat**
5: συγχώνευση των δυο «κοντινότερων» κλάσεων
6: επανάληψη υπολογισμού της μέσης «απόστασης» για κάθε κλάση C_i και C_j όπου είναι ο μέση «απόσταση» μεταξύ οποιουδήποτε αντικειμένου ανήκει στη κλάση C_i και οποιουδήποτε αντικειμένου ανήκει στη κλάση C_j
7: **until** μέχρι να παραμείνει μόνο μια μοναδική κλάση
8: **return** $C_1, C_2, \cdots, C_c$
9: Τέλος

Clustering Tendency

-Έξοδος αρχείου ομαδοποίησης του αλγορίθμου GAAC (η δεξιά στήλη αντιστοιχεί στον κωδικό-όνομα κάθε κλάσης):

majority, spokesman, spokeswoman 4
east, italian, asian, arkansas 48
cents, billion, sixteenths, point, thousand, minutes, million, percent, basis, points 242
over, about, else 247
friday, monday, thursday, tuesday, wednesday 265
contributed, structure, enterprises, motors, recession, bet, valley, commitments 482
⋮
<unk> 510
</s> 511
<s> 512

4.6 Ομαδοποίηση με χρήση του εργαλείου Cluto

4.6.1 Τι είναι το Cluto

Το CLUTO είναι ένα πακέτο λογισμικού για ομαδοποίηση συνόλου δεδομένων μικρής και μεγάλης διάστασης καθώς και για την ανάλυση των χαρακτηριστικών των διαφόρων κλάσεων. Στο κεφάλαιο αυτό, καθώς και στα επόμενα τέσσερα, θα προσπαθήσουμε να περιγράψουμε στοιχεία του εργαλείου τα οποία χρησιμοποιήσαμε, για περαιτέρω πληροφορίες υπάρχει διαθέσιμο manual οδηγιών [15].

Το CLUTO παρέχει τρεις διαφορετικές κατηγορίες αλγορίθμων ομαδοποίησης, που λειτουργούν είτε απευθείας στο χώρο χαρακτηριστικών του αντικειμένου ή στο χώρο της ομοιότητας του αντικειμένου. Οι αλγόριθμοι αυτοί βασίζονται στις partitional (διαμερισμένες/διχοτομημένες), agglomerative (συσσωρευτικές) και graph-partitioning (γραφικά διαχωρισμένες) τεχνικές ομαδοποίησης. Ένα βασικό χαρακτηριστικό στους περισσότερους αλγόριθμους ομαδοποίησης του CLUTO είναι ότι αντιμετωπίζουν το πρόβλημα της ομαδοποίησης ως μια διαδικασία βελτιστοποίησης η οποία επιδιώκει να μεγιστοποιεί ή να ελαχιστοποιεί ένα συγκεκριμένο clustering criterion function (κριτήριο συνάρτησης της ομαδοποίησης), καθορισμένο είτε γενικά είτε τοπικά, πάνω στο χώρο λύσης του clustering. Το CLUTO πα-

ρέχει ένα σύνολο επτά διαφορετικών criterion functions που μπορούν να χρησιμοποιηθούν για να οδηγήσουν τόσο σε partitional όσο και σε agglomerative αλγόριθμους ομαδοποίησης. Οι περισσότερες από αυτές τις criterion functions έχει αποδειχθεί, ότι παράγουν υψηλής ποιότητας λύσεις clustering σε σύνολα δεδομένων υψηλής διάστασης, ιδίως εκείνων που προκύπτουν από το clustering κειμένων. Εκτός από αυτές τις criterion functions, το CLUTO παρέχει μερικά από τα πιο «παραδοσιακά» τοπικά κριτήρια (local criteria) (π.χ. single-link, complete-link and UPGMA), τα οποία μπορούν να χρησιμοποιηθούν στο πλαίσιο του agglomerative clustering. Επιπλέον , το CLUTO παρέχει graph-partitioning-based αλγορίθμους ομαδοποίησης που είναι κατάλληλοι για την εύρεση κλάσεων που συγκροτούν γειτονικές περιοχές που εκτείνονται σε διαφορετικές διαστάσεις του υποκείμενου (underlying) χώρου χαρακτηριστικών.

Μια σημαντική πτυχή του partitional-based criterion με γνώμονα αλγόριθμους ομαδοποίησης είναι η μέθοδος που χρησιμοποιείται για τη βελτιστοποίηση αυτού του criterion functions. Το CLUTO χρησιμοποιεί έναν «τυχαιοποιημένο» αυξητικό (incremental) αλγόριθμο βελτιστοποίησης, που είναι άπληστος, έχει χαμηλές υπολογιστικές απαιτήσεις και έχει αποδειχθεί ότι παράγει υψηλής ποιότητας λύσεων clustering. Οι εφαρμογές του CLUTO για graph-partitioning-based αλγόριθμους ομαδοποίησης χρησιμοποιούν υψηλής ποιότητας και αποτελεσματικής πολυεπίπεδης διαμέριση γράφων (multilevel graph partitioning) αλγορίθμους, οι οποίοι προέρχονται από METIS και HMETIS γράφους και hypergraph(υπεργράφους) partitioning αλγορίθμους.

Το CLUTO παρέχει επίσης εργαλεία για την ανάλυση κλάσεων που έχουν δημιουργηθεί, με σκοπό την κατανόηση των σχέσεων μεταξύ των αντικειμένων που έχουν ανατεθεί σε κάθε κλάση και των σχέσεων μεταξύ των διαφορετικών κλάσεων, και εργαλεία για την οπτικοποίηση των ομαδοποιήσεων που έχουν πραγματοποιηθεί. Το CLUTO μπορεί να προσδιορίσει τα χαρακτηριστικά που περιγράφουν καλύτερα ή/και διακρίνουν κάθε κλάση. Αυτά τα σύνολα χαρακτηριστικών μπορούν να χρησιμοποιηθούν για να αποκτήσουμε μια καλύτερη κατανόηση του συνόλου των αντικειμένων που έχουν ανατεθεί σε κάθε cluster και στο να παρέχει συνοπτικές περιλήψεις σχετικά με το περιεχόμενο των clusters. Επιπλέον, το CLUTO παρέχει δυνατότητες οπτικοποίησης που μπορούν να χρησιμοποιηθούν για να παρατηρήσουμε τις σχέσεις μεταξύ των clusters, των αντικειμένων και των χαρακτηριστικών.

Οι αλγόριθμοι του CLUTO έχουν βελτιστοποιηθεί για τη λειτουργία σε πολύ μεγάλα σύνολα δεδομένων τόσο όσον αφορά τον αριθμό των αντικειμένων, καθώς και τον αριθμό των διαστάσεων. Αυτό ισχύει ιδιαίτερα για τους αλγορίθμους του CLUTO για partitional ομαδοποίηση. Αυτοί οι αλγόριθμοι μπορούν γρήγορα να ομαδοποιήσουν σύνολα δεδομένων

με πολλές δεκάδες χιλιάδες αντικείμενα και αρκετές χιλιάδες διαστάσεις. Επιπλέον, δεδομένου ότι τα περισσότερα μεγάλων διαστάσεων, σύνολα δεδομένων είναι πολύ αραιά (sparse), το CLUTO λαμβάνει άμεσα υπόψη αυτήν την αραιότητα (sparsity) και απαιτεί μνήμη που είναι σχεδόν γραμμική για το μέγεθος της εισόδου. Αυτή τη πρόβλεψη που μας παρέχει το CLUTO δεν την εκμεταλλευτήκαμε στη δική μας περίπτωση, καθώς τα δεδομένα μας είχαν συμπιεστεί με την τεχνική SVD με αποτέλεσμα να είναι πλήρως πυκνά (dense).

Το CLUTO αποτελείται από δύο αυτόνομα προγράμματα (vcluster και scluster) για την ομαδοποίηση και την ανάλυση των κλάσεων, καθώς και μια βιβλιοθήκη μέσω της οποίας ένα πρόγραμμα εφαρμογής μπορεί να έχει άμεση πρόσβαση στους διάφορους αλγόριθμους ομαδοποίησης και ανάλυσης που εφαρμόζονται στο CLUTO.

Σχετικά με τους αλγορίθμους που περιέχει το CLUTO, οι partitional (or divisive) αλγόριθμοι ομαδοποίησης είναι ο rb, ο rbr και ο direct. Οι agglomerative αλγόριθμοι ομαδοποίησης είναι ο agglo και ο bagglo. Τέλος, ο graph-partitioning αλγόριθμος που περιλαμβάνεται είναι ο graph. Στην εργασία μας, έμεσα χρησιμοποιήσαμε όλους τους αλγορίθμους ομαδοποίησης, στην πράξη όμως δεν χρησιμοποιήσαμε τους αλγορίθμους rb και agglo οι οποίοι όμως, αποτελούν κομμάτι της υλοποίησης των αλγορίθμων rbr και bagglo αντίστοιχα. Όλες τις παραπάνω εφαρμογές, τις αναλύουμε εκτενέστερα στα κεφάλια που ακολουθούν.

4.6.2 Rbr Αλγόριθμος

Η τεχνική ομαδοποίησης rb αποτελεί κομμάτι της υλοποίησης για τον αλγόριθμο rbr επομένως, θεωρούμε σοφότερο να αναλύσουμε αρχικά τον τρόπο υλοποίησής του. Πιο συγκεκριμένα, στην μέθοδο αυτή, η επιθυμητή k-way ομαδοποίηση υπολογίζεται εκτελώντας μια ακολουθία από $k-1$ επαναλαμβανόμενες διχοτομήσεις (bisections). Στην προσέγγιση αυτή, ο πίνακας των διανυσμάτων αρχικά, ομαδοποιείται σε δύο κλάσεις, στη συνέχεια, μία από αυτές τις κλάσεις επιλέγεται και χωρίζεται περαιτέρω σε δύο κλάσεις. Αυτή η διαδικασία συνεχίζεται μέχρις ότου βρεθεί ο επιθυμητός αριθμός κλάσεων. Κατά τη διάρκεια κάθε βήματος, η κάθε κλάση διχοτομείται έτσι ώστε, η προκύπτουσα 2-way ομαδοποίηση να βελτιστοποιεί μια συγκεκριμένη clustering criterion function (η οποία επιλέγεται χρησιμοποιώντας τη -crfun παράμετρο). Σημειώνουμε ότι αυτή η προσέγγιση εξασφαλίζει ότι η criterion function είναι τοπικά βελτιστοποιημένη σε κάθε διχοτόμηση, αλλά σε γενικές γραμμές η ομαδοποίηση δεν έχει βελτιστοποιηθεί καθολικά.

Η τεχνική ομαδοποίησης rbr είναι ο αλγόριθμος που υλοποιήσαμε στην δική μας εργα-

σία. Αναλυτικότερα, σε αυτή τη μέθοδο η επιθυμητή k-way ομαδοποίηση υπολογίζεται με τρόπο παρόμοιο με την επαναλαμβανόμενη-διχοτόμηση του αλγορίθμου rb, με τη διαφορά ότι στο τέλος, η συνολική λύση της ομαδοποίησης είναι καθολικά βελτιστοποιημένη (repeated-bisecting). Ουσιαστικά , η εντολή vcluster χρησιμοποιεί τη λύση που λαμβάνεται, ως αρχική λύση, από την ομαδοποίηση rb και προσπαθεί να βελτιστοποιήσει περαιτέρω το clustering criterion function.

Όπως έχουμε ήδη αναφερθεί, το CLUTO παρέχει εργαλεία για την ανάλυση κλάσεων που έχουν δημιουργηθεί περιέχοντας σημαντικές πληροφορίες σχετικά με την ομαδοποίηση που λαμβάνει χώρα. Πιο συγκεκριμένα, το σχετικό αρχείο εξόδου έχει την εξής μορφή:

****

vcluster (CLUTO 2.1.2) Copyright 2001-06, Regents of the University of Minnesota

Matrix Information ——————————————————

$Name : dense\_matrix$, $\sharp Rows$: 2700, $\sharp Columns$: 100, $\sharp NonZeros$: 270000

Options ——————————————————————-

CLMethod=RBR, CRfun=I2, SimFun=CorrCoef, $\sharp Clusters = 512$

RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40

Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5

CSType=Best, AggloFrom=0, AggloCRFun=I2, NTrials=20, NIter=20

Solution ———————————————————————

———————————————————————

512-way clustering: [I2=2.10e+03] [2700 of 2700]

———————————————————————

cid	Size	ISim	ISdev	ESim	ESdev	
0	2	+0.960	+0.000	+0.011	+0.005	
1	2	+0.916	+0.000	+0.000	+0.000	
2	7	+0.887	+0.012	+0.007	+0.002	
3	2	+0.877	+0.000	+0.002	+0.001	

⋮

511 6 +0.483 +0.030 +0.036 +0.008 |

Timing Information ——————————————————
I/O: 0.144 sec
Clustering: 17.828 sec
Reporting: 0.012 sec
Memory Usage Information ————————————————
Maximum memory used: 5808128 bytes
Current memory used: 2339096 bytes
****

Στη κατηγορία options του αρχείου περιέχεται το σύνολο των παραμέτρων που επιλέχθηκαν για την εκτέλεση της ομαδοποίησης. Στις συγκεκριμένες παραμέτρους καταλήξαμε, τόσο μέσα από παρατηρήσεις στα δεδομένα μας, όσο και από μια σειρά συγκρίσεων μεταξύ των αποτελεσμάτων του μοντέλου για διαφορετικές παραμέτρους (αυτό θα γίνει πιο κατανοητό στη κεφάλαιο 5). Αναλυτικότερα, οι παράμετροι που προσδιορίσαμε είναι οι εξής [1]:

-CLMethod=RBR: cluster method, επιλέγετε η μέθοδο ομαδοποίησης για τα αντικείμενα. Στη προκειμένη περίπτωση επιλέγεται η μέθοδος ομαδοποίησης rbr
-CRfun=I2: citerion function, επιλέγεται το κριτήριο συνάρτησης της ομαδοποίησης. Στη προκειμένη περίπτωση επιλέγεται το κριτήριο $I2 = maximize \sum_{i=1}^{k} \sqrt{\sum_{v,u \in S_i} sim(v,u)}$ το οποίο επιλέχτηκε και στη μέθοδο ομαδοποίησης direct. Ο λόγος επιλογής αυτού του κριτηρίου, οφείλεται στο γεγονός ότι (σύμφωνα με το CLUTO) αποτελεί την επιλογή που επιφέρει τα καλύτερα αποτελέσματα σε αυτού του είδους τις ομαδοποιήσεις και στην αδυναμία εκτέλεσης τόσων διαφορετικών πειραμάτων (υπάρχουν διαθέσιμα εφτά διαφορετικά κριτήρια).
-SimFun=CorrCoef: similarity function, επιλέγεται η συνάρτηση ομοιότητας που χρησιμοποιείται για τα διανύσματα λέξεων. Σε αυτή τη περίπτωση επιλέγεται η συνάρτηση cosine λόγο των καλύτερων αποτελεσμάτων του μοντέλου, σε σχέση με τις υπόλοιπες συναρτήσεις.
-Clusters=512: επιλέγεται ο επιθυμητός αριθμός κλάσεων
-CSType=Best: επιλέγεται η μέθοδος που χρησιμοποιείται για τα επιλέξει την επόμενη κλά-

[1] οι παράμετροι που θεωρήσαμε λιγότερο σημαντικοί ή δεν επηρέαζαν το αποτέλεσμα της ομαδοποίησης, λαμβάνουν τις default τιμές που καθορίζει το CLUTO

ση που θα διχοτομηθεί (bisect). Στη συγκεκριμένη περίπτωση επιλέγεται η μέθοδος Best, καθώς είναι η κατάλληλη για τα δεδομένα μας (πυκνά δεδομένα, κλάσεις με μοιρασμένο αριθμό αντικειμένων)
-NTrials=20: επιλέγεται ο αριθμός των διαφορετικών λύσεων ομαδοποίησης που θα υπολογιστούν για τον αλγόριθμος που έχουμε επιλέξει. Στη συνέχεια, διατηρείται η λύση που έχει τη καλύτερη τιμή για το κριτήριο συνάρτησης που έχουμε εφαρμόσει. Επιλέξαμε τον υπολογισμό είκοσι διαφορετικών λύσεων για ακόμα καλύτερα αποτελέσματα ομαδοποίησης.
-NIter=20: επιλέγεται ο μέγιστος αριθμός των επαναλήψεων βελτίωσης που πρέπει να εκτελεστούν, σε κάθε βήμα του clustering. Επιλέξαμε τον υπολογισμό είκοσι για ακόμα καλύτερα αποτελέσματα ομαδοποίησης.

Clustering Tendency
-Έξοδος αρχείου ομαδοποίησης του αλγορίθμου rbr (η δεξιά στήλη αντιστοιχεί στον κωδικόόνομα κάθε κλάσης):

majority, spokesman, spokeswoman 4
east, italian, asian, arkansas 48
cents, billion, sixteenths, point, thousand, minutes, million, percent, basis, points 242
over, about, else 247
friday, monday, thursday, tuesday, wednesday 265
contributed, structure, enterprises, motors, recession, bet, valley, commitments 482
⋮
<unk> 510
</s> 511
<s> 512

4.6.3 Direct Αλγόριθμος

Στη μέθοδο αυτή, η επιθυμητή k-way ομαδοποίηση υπολογίζεται με την ταυτόχρονη εύρεση όλων των κλάσεων k. Σε γενικές γραμμές, εκτιμώντας μια k-way ομαδοποίηση άμεσα (directly) είναι πιο αργή απ΄ ό,τι η ομαδοποίηση μέσω επαναληπτικών bisections που περιγράψαμε στο προηγούμενο κεφάλαιο. Από την οπτική που αφορά τη ποιότητα, για λογικά μικρές τιμές του k (συνήθως μικρότερο από 10-20), η άμεση (direct) προσέγγιση οδηγεί

σε καλύτερες ομαδοποιήσεις των διανυσμάτων, σε σχέση μ΄ εκείνες που λαμβάνονται μέσω επανειλημμένων bisections. Ωστόσο, καθώς αυξάνεται το k, η προσέγγιση της επαναλαμβανόμενης-διχοτόμησης τείνει να είναι καλύτερη από το direct clustering.

Το CLUTO παρέχει εργαλεία για την ανάλυση κλάσεων που έχουν δημιουργηθεί περιέχοντας σημαντικές πληροφορίες σχετικά με την ομαδοποίηση που λαμβάνει χώρα. Πιο συγκεκριμένα, το σχετικό αρχείο εξόδου έχει την εξής μορφή:

```
*******************************************************************************
vcluster (CLUTO 2.1.2) Copyright 2001-06, Regents of the University of Minnesota

Matrix Information ------------------------------------------------------
Name: dense_matrix, #Rows: 2700, #Columns: 100, #NonZeros: 270000

Options -----------------------------------------------------------------
CLMethod=Direct, CRfun=I2, SimFun=CorrCoef [1], #Clusters = 512
RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
CSType=Best, AggloFrom=0, AggloCRFun=I2, NTrials=20, NIter=20

Solution ----------------------------------------------------------------

------------------------------------------------------------------------
512-way clustering: [I2=2.11e+03] [2700 of 2700]
------------------------------------------------------------------------
cid Size ISim ISdev ESim ESdev |
------------------------------------------------------------------------
0   2  +0.916  +0.000  +0.001  +0.000  |
1   4  +0.900  +0.019  +0.008  +0.003  |
2   7  +0.890  +0.012  +0.007  +0.002  |
3   2  +0.878  +0.000  +0.004  +0.004  |
⋮
511 6  +0.465  +0.055  +0.013  +0.004  |
------------------------------------------------------------------------
```

[1] η ομοιότητα μεταξύ των αντικειμένων υπολογίζεται χρησιμοποιώντας την correlation coefficient

Timing Information ——————————————————
I/O: 0.144 sec
Clustering: 208.976 sec
Reporting: 0.008 sec
Memory Usage Information ————————————————
Maximum memory used: 4841472 bytes
Current memory used: 2339096 bytes
****

Το σύνολο των παραμέτρων και σε αυτήν την περίπτωση, επιλέχτηκε με την αντίστοιχη τακτική με την προηγούμενη μέθοδο ομαδοποίησης. Η σημαντικότερη διαφοροποίηση, εκτός από την μέθοδο ομαδοποίησης, αφορά τη similarity function που χρησιμοποιήθηκε.

-Έξοδος αρχείου ομαδοποίησης του αλγορίθμου direct (η δεξιά στήλη αντιστοιχεί στον κωδικό-όνομα κάθε κλάσης):

considering, hoping, facing, planning 4
youre, im, everybody, theyre, everyone, weve, ive, theres, id, theyve, lets 5
friday, monday, thursday, tuesday, wednesday 113
phone, radio, training, communications, highway, shopping, television 377
⋮
<unk> 513
</s> 514
<s> 515

4.6.4 Bagglo Αλγόριθμος

Η τεχνική ομαδοποίησης agglo αποτελεί κομμάτι της υλοποίησης για τον αλγόριθμο bagglo επομένως, θεωρούμε σοφότερο να αναλύσουμε αρχικά, τον τρόπο υλοποίησης του. Πιο συγκεκριμένα, στη μέθοδο αυτή, η επιθυμητή k-way ομαδοποίηση υπολογίζεται χρησιμοποιώντας την agglomerative μεθοδολογία, που έχουμε ήδη περιγράψει στα παραπάνω κεφάλαια, της οποίας στόχος είναι να βελτιστοποιεί τοπικά (ελαχιστοποίηση ή μεγιστοποίηση)

μίας συγκεκριμένη clustering criterion function (η οποία επιλέγεται χρησιμοποιώντας την παράμετρο -crfun). Η λύση της ομαδοποίησης λαμβάνεται με τη διακοπή της συσσωρευτικής (agglomerative) διαδικασίας όταν απομένουν k κλάσεις.

Η τεχνική ομαδοποίησης bagglo είναι ο αλγόριθμος που υλοποιήσαμε στην δουλειά μας. Αναλυτικότερα, σε αυτή τη μέθοδο η επιθυμητή k-way ομαδοποίηση υπολογίζεται με τρόπο παρόμοιο με τη μέθοδο agglo ωστόσο, η διαδικασία συσσώρευσης ωθείται από μια partitional ομαδοποίηση που αρχικώς υπολογίζεται επί του συνόλου δεδομένων. Όταν επιλέγεται ο αλγόριθμος bagglo αρχικά, το CLUTO υπολογίζει μια $\sqrt{n}$-way ομαδοποίηση χρησιμοποιώντας τη τεχνική rb, όπου n είναι ο αριθμός των διανυσμάτων-λέξεων που πρέπει να ομαδοποιηθούν. Στη συνέχεια, αυξάνεται το αρχικό διάστημα των χαρακτηριστικών με την προσθήκη $\sqrt{n}$ νέων διαστάσεων, μια για κάθε κλάση. Επιπλέον, σε κάθε αντικείμενο αντιστοιχείται μια τιμή από τη διάσταση, αντίστοιχη με την κλάση στην οποία ανήκει, όπου η τιμή αυτή είναι ανάλογη με την ομοιότητα μεταξύ του εν λόγω αντικειμένου και του κεντροειδούς της κλάσης (cluster-centroid). Καταλήγοντας, με δεδομένη αυτή την επαυξημένη εκπροσώπηση, η συνολική λύση της ομαδοποίησης επιτυγχάνεται χρησιμοποιώντας την παραδοσιακή agglomerative μεθοδολογία. Η επιθυμητή ομαδοποίηση λαμβάνεται με τη διακοπή της διαδικασίας όταν έχουν απομένει k κλάσεις. Τα πειράματά τα οποία πραγματοποιήσαμε στα μοντέλα μας, έδειξαν ότι αυτή η εναλλακτική agglomerative προσέγγιση ξεπερνούσε πάντα την απόδοση του παραδοσιακού agglomerative αλγορίθμου.

Το CLUTO παρέχει εργαλεία για την ανάλυση κλάσεων που έχουν δημιουργηθεί περιέχοντας σημαντικές πληροφορίες σχετικά με την ομαδοποίηση που λαμβάνει χώρα. Πιο συγκεκριμένα, το σχετικό αρχείο εξόδου έχει την εξής μορφή:

```
**********************************************************************
vcluster (CLUTO 2.1.2) Copyright 2001-06, Regents of the University of Minnesota

Matrix Information ------------------------------------------------------
Name : dense_matrix, #Rows : 2700, #Columns : 100, #NonZeros : 270000

Options -----------------------------------------------------------------
CLMethod=BAGGLO, CRfun=CLINK [1], SimFun=CorrCoef [2], #Clusters : 512
RowModel=None, ColModel=None, GrModel=SY-DIR, NNbrs=40
```

[1] επιλέγεται κριτήριο συνάρτησης πλήρους δεσμού (complete link)

[2] η ομοιότητα μεταξύ των αντικειμένων υπολογίζεται χρησιμοποιώντας την correlation coefficient

Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
CSType=Best, AggloFrom=0, AggloCRFun=CLINK, NTrials=20, NIter=20

Solution ———————————————————

512-way clustering: [CLINK=0.00e+00] [2700 of 2700]

---

cid	Size	ISim	ISdev	ESim	ESdev	
0	3	+0.609	+0.037	+0.017	+0.010	
1	6	+0.520	+0.058	+0.016	+0.009	
2	9	+0.438	+0.037	+0.023	+0.006	
3	2	+0.635	+0.000	+0.029	+0.001	
⋮						
511	3	+0.583	+0.022	+0.010	+0.007	

Timing Information ———————————————
I/O: 0.144 sec
Clustering: 7.164 sec
Reporting: 0.012 sec
Memory Usage Information ———————————————
Maximum memory used: 75968512 bytes
Current memory used: 2360752 bytes
****

Το σύνολο των παραμέτρων και σε αυτήν την περίπτωση, επιλέχτηκε με την αντίστοιχη τακτική με τις προηγούμενες μεθόδους ομαδοποίησης. Η σημαντικότερη διαφοροποίηση, εκτός από την μέθοδο ομαδοποίησης, αφορά τη criterion function και τη similarity function που χρησιμοποιήθηκαν.

-Έξοδος αρχείου ομαδοποίησης του αλγορίθμου bagglo (η δεξιά στήλη αντιστοιχεί στον κωδικό-όνομα κάθε κλάσης):

loss, takeover, dividend 1
finally, essentially, certainly, now, then, clearly 2
health, medical, insurance, security 9
friday, monday, sunday, thursday, tuesday, wednesday 360
japan, beijing, light, frankfurt, india, washington, vietnam, britain, israel, south, asia, germany, korea, europe, tokyo, russia, mexico, china, eastern, france, bosnia 394
fixed, global 420
⋮
<unk> 513
</s> 514
<s> 5152

4.6.5 Graph Αλγόριθμος

Στη μέθοδο αυτή, η επιθυμητή k-way ομαδοποίηση υπολογίζεται από την αρχική μοντελοποίηση των αντικειμένων, χρησιμοποιώντας ένα γράφημα (graph) πλησιέστερου γείτονα (κάθε αντικείμενο/διάνυσμα λέξης γίνεται μια κορυφή, και στη συνέχεια συνδέεται με τα πιο παρόμοια με αυτό αντικείμενα), και κατόπιν «κόβει» το γράφημα σε k-clusters χρησιμοποιώντας ένα minimum-cut graph[1] αλγόριθμο διαχωρισμού. Σημειώνουμε ότι εάν το γράφημα περιέχει περισσότερα από ένα συνδεδεμένα components («κομμάτια»), τότε ο αλγόριθμος επιστρέφει μια (k+m)-way ομαδοποίηση, όπου m είναι ο αριθμός των συνδεδεμένων components στο γράφημα.

Το CLUTO παρέχει εργαλεία για την ανάλυση κλάσεων που έχουν δημιουργηθεί περιέχοντας σημαντικές πληροφορίες σχετικά με την ομαδοποίηση που λαμβάνει χώρα. Πιο συγκεκριμένα, το σχετικό αρχείο εξόδου έχει την εξής μορφή:

****
vcluster (CLUTO 2.1.2) Copyright 2001-06, Regents of the University of Minnesota

Matrix Information ----------

[1] Στη θεωρία γράφων, ένας minimum cut graph είναι μια περικοπή, όπου κόβει το σύνολο το οποίο έχει το μικρότερο αριθμό ακμών (μη σταθμισμένη περίπτωση) ή το μικρότερο δυνατό άθροισμα βαρών.

$Name$: $dense\_matrix$, ♯$Rows$: 2700, ♯$Columns$: 100, ♯$NonZeros$: 270000

Options ————————————————————————-
CLMethod=GRAPH, CRfun=Cut [1], SimFun=EuclDist [2], ♯$Clusters$ = 512
RowModel=None, ColModel=None, GrModel=ASY-DIR, NNbrs=40
Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
CSType=Best, AggloFrom=0, AggloCRFun=SLINK_W, NTrials=20, NIter=20

Solution ————————————————————————

———————————————————————————

512-way clustering: [Cut=9.30e+04] [2700 of 2700]

———————————————————————————

cid Size ISim ISdev ESim ESdev |

———————————————————————————

0 12 +0.946 +0.026 +0.016 +0.006 |
1 8 +0.989 +0.001 +0.014 +0.001 |
2 7 +0.995 +0.001 +0.015 +0.002 |
3 7 +0.990 +0.002 +0.014 +0.001 |
⋮
511 1 +nan +nan +0.075 +nan |

———————————————————————————

Timing Information ————————————————
I/O: 0.144 sec
Clustering: 13.900 sec
Reporting: 0.008 sec
Memory Usage Information ———————————————
Maximum memory used: 14680064 bytes
Current memory used: 4006400 bytes
****

[1] επιλέγεται το κριτήριο συνάρτησης (cut)
[2] η ομοιότητα μεταξύ των αντικειμένων υπολογίζεται χρησιμοποιώντας την Ευκλείδεια απόσταση

Το σύνολο των παραμέτρων και σε αυτήν την περίπτωση, επιλέχτηκε με την αντίστοιχη τακτική με την προηγούμενες μεθόδους ομαδοποίησης. Η σημαντικότερη διαφοροποίηση, εκτός από την μέθοδο ομαδοποίησης, αφορά τη criterion function και τη similarity function που χρησιμοποιήθηκαν.

-Έξοδος αρχείου ομαδοποίησης του αλγορίθμου graph (η δεξιά στήλη αντιστοιχεί στον κωδικό-όνομα κάθε κλάσης):

cents, billion, point, thousand, pages, seconds, million, trillion, pence, basis, cent, points 1
successful, popular, costly, heavily, crucial, critical, profitable 3
them, us, him, me 4
hunter, lynch,,fuel, francisco, angeles, metal 32
yesterday, friday, monday, tuesday, wednesday 484
restrictions, transactions, events, charges, lawsuits, suits 489
⋮
<unk> 513
</s> 514
<s> 515

Κεφάλαιο 5

Βελτιστοποίηση και Πειραματική Αξιολόγηση

5.1 Περιγραφή των δεδομένων

Σχήμα 5.1: Wall Street Journal

5. ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΚΑΙ ΠΕΙΡΑΜΑΤΙΚΗ ΑΞΙΟΛΟΓΗΣΗ

Για την υλοποίηση του πειραματικού κομματιού της διπλωματικής, χρησιμοποιούμε τα κείμενα από την Wall Street Journal σαν δεδομένα εκπαίδευσης και δοκιμής (train and test data). Ειδικότερα, χρησιμοποιήσαμε άρθρα από το 1994. Τα αρχεία δεδομένων χρησιμοποιούν το συμβολικό όνομα WS94_*VPZ και κάθε πρόταση έχει την ακόλουθη μορφή:

<p.wsj94_001.0012.3>
<s.wsj94_001.0012.3.2>

D.D.T. is a highly persistent chemical that moves up the food chain, COMMA and it accumulates in the fatty tissue of humans .PERIOD
</s>
</p>

Η πρώτη γραμμή σημαίνει ότι η πρόταση προέρχεται από το έγγραφο 001.0012, παράγραφος 3, ενώ η δεύτερη σειρά υποδηλώνει ότι αναφερόμαστε σε πρόταση. Τα δεδομένα του κειμένου δεν έχουν κανονικοποιηθεί, η μόνη επέμβαση που κάναμε ήταν να εκφράσουμε τα πάντα σε λέξεις (verbalize). Πιο συγκεκριμένα, τα σημεία στίξης αντικαθίστανται από τη λέξη του σημείου, (π.χ. , ⇒ comma). Επιπλέον, υπάρχουν πολλές λανθασμένες λέξεις και έτσι ήταν απαραίτητο να επεξεργαστούμε τα δεδομένα πριν από τη διαίρεση τους σε train και test data. Το σύνολο των επεμβάσεων που πραγματοποιήθηκαν στα κείμενα είναι οι εξής:

- Αρχικά, αφαιρούμε όλα τις επικεφαλίδες, όπως <p.wsj94_001.0012.3> και <s.wsj94_001.0012.3.2>
- Μετατρέπουμε όλα τα δεδομένα του κειμένου σε πεζά
- Απορρίπτουμε ανακριβείς λέξεις όπως "kknow", "tthey", "aare"
- Απομακρύνουμε ψηφία και αντικαθιστούμε τους αριθμούς με τις αντίστοιχες λέξεις τους
- Η στίξη αναπαριστάται μόνο από τη λέξη καθώς αφαιρούμε τα σημεία που αφορούν τα σημεία στίξης
- Προσθέτουμε τους χαρακτήρες <s> και </s> που υποδηλώνουν, ο μεν πρώτος την αρχή και μεν δεύτερος το τέλος κάθε πρότασης.
- Για το μικρό λεξιλόγιο, βρίσκουμε τις 2700 πιο συχνές λέξεις οι οποίες αποτελέσαν και το λεξιλόγιο μας στη περίπτωση αυτή, και αντικαθιστούμε, οποιαδήποτε από τις λέξεις βρίσκονται εκτός λεξιλογίου με το σύμβολο unk. Στο μεγάλο λεξιλόγιο μας,

δεν αντικαθιστούμε καμία από τις λέξεις και χρησιμοποιούμε όλες τις διαφορετικές λέξεις που εμφανίζονται στα δεδομένα.

- Σύμφωνα με το λεξιλόγιο, αναπαραστήσαμε κάθε λέξη με τη χρήση μιας αλληλουχία δεικτών. Κάθε δείκτης δείχνει τη θέση κάθε λέξης, δεδομένου ότι τις έχουμε ταξινομήσει κατά φθίνουσα σειρά ξεκινώντας από την λέξη με την μεγαλύτερη συχνότητα. Για παράδειγμα έχουμε τη πρόταση:

  <s> like d d t itself comma <unk> about the nineteen seventy two decision to <unk> certain uses of the chemical are <unk> <unk> period </s>

  η οποία μετατρέπεται σε:

  4 156 178 178 125 829 2 1 64 3 40 128 20 508 7 1 656 1963 8 3 1463 36 1 1 6 5

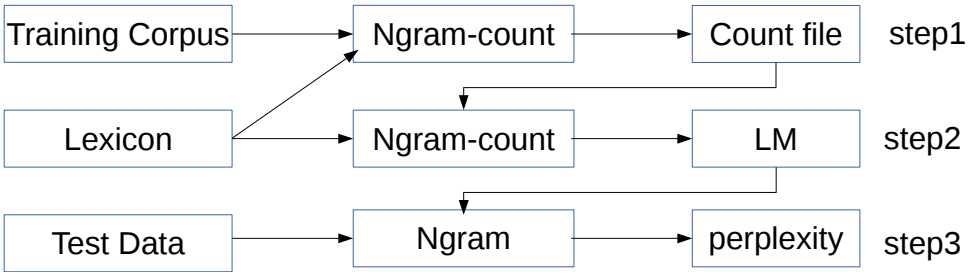
- Έτσι καταλήγοντας, τα train και test data περιλαμβάνουν τα στοιχεία που απεικονίζονται στο πίνακα 5.1:

WS94	Train data	Test data
No of sentences	150000	6000
No of words	4169233	164570

Πίνακας 5.1: Περιγραφή δεδομένων

5.2 Basiline expiriments

Στο κεφάλαιο 4.3 περιγράψαμε το εργαλείο SRILM. Στο συγκεκριμένο στάδιο, χρησιμοποιούμε το εργαλείο με σκοπό την παραγωγή ενός γλωσσικού μοντέλου στο διακριτό χώρο, βασισμένο πάνω σε trigram. Κατά την εκτέλεση των σταδίων του εργαλείου, αρχικά δημιουργούμε το n-gram count file για το corpus, στη συνέχεια εκπαιδεύουμε το γλωσσικό μοντέλο κάνοντας χρήση του n-gram count file και υπολογίζουμε το test data perplexity(5.2).



Σχήμα 5.2: SRILM toolkit

Η εκτέλεση των εντολών του εργαλείου SRILM για την εξαγωγή των αποτελεσμάτων έγινε μέσω της γραμμής εντολών του Linux και αποτελείται από 3 κύρια στάδια:

- *Πρώτο στάδιο*: Δημιουργεί το n-gram count file από το corpus και κάνει εκτίμηση του γλωσσικού μοντέλου από αυτά.

$$ngram - count - vocab\ Lexicon.file \\ - text\ train.txt \\ - order\ 3 \\ - write\ train\_3gram \\ - unk$$

Παράμετροι:
-vocab: διαβάζει το λεξιλόγιο από το αρχείο που δίνεται
-text: εκπαιδεύει το corpus που δίνεται
-order: θέτει το μέγιστο μέγεθος των N-grams
-write: το αρχείο εξόδου που περιέχει τα N-gram counts
-unk: θέτει κάθε λέξη που βρίσκεται εκτός του λεξιλογίου (OOV-Out Of Vocabulary) ως <unk>

-Έξοδος αρχείου train_3gram

$< unk >\ 516381$
$< unk >< unk >\ 75949$
$< unk >< unk >< unk >\ 9531$
$< unk >< unk >\ sound\ 7$

$< unk > < unk >\ martin$ 3
⋮
$quite$ 236
$quite\ < unk >$ 70
$quite\ < unk > < unk >$ 4
$quite\ < unk >\ up$ 1
$quite\ < unk >\ comma$ 17
⋮

- *Δεύτερο στάδιο*: Διαβάζει το αρχείο που περιέχει τα n-gram counts και εκπαιδεύει το γλωσσικό μοντέλο.

  Κάθε πίνακας εκπαίδευσης n-grams είναι αραιός για μεγάλα σώματα κειμένων (νόμος του Zipf). Στο στάδιο αυτό, μια πολύ συχνά χρησιμοποιημένη βελτιστοποίηση, είναι η εξομάλυνση smoothing πάνω στη συνάρτηση των λέξεων w, $P(w|h)$ για κάθε ιστορικό h, και δίνοντας σε κάθε λέξη w με πιθανότητα $P(w|h) = 0$ μια μικρή τιμή μεγαλύτερη από 0. Οι τεχνικές smoothing αντιμετωπίζουν τα προβλήματα που εμφανίζονται με λέξεις που δεν έχουμε συναντήσει σε ένα σώμα κειμένων.

  Από που όμως προέρχονται αυτές οι μικρές τιμές; Η απάντηση είναι, μειώνοντας τις πιθανότητες $P(w|h)$ των λέξεων w των οποίων οι πιθανότητες είναι μεγαλύτερες από μηδέν, $P(w|h) > 0$. Αυτό το στάδιο είναι γνωστό και ως discounting. Στη συνέχεια, οι μικρές τιμές που έχουν συγκεντρωθεί, πρέπει να κατανεμηθούν σε αυτές τις λέξεις, με πιθανότητα $P(w|h)$. Αυτή η διαδικασία discounting, βασισμένη σε διάφορες τεχνικές smoothing μπορεί να παρομοιαστεί σαν να φορολογείς πλούσιους ανθρώπους και να μοιράζεις τα χρήματα αυτά στους φτωχούς, αυτούς που έχουν 0 cents

  Όπως έχουμε ήδη αναφέρει, υπάρχουν διάφορες τεχνικές smoothing, τις οποίες δοκιμάζουμε πάνω στα δεδομένα δοκιμής, ώστε να βρούμε την μεγαλύτερη ακρίβεια για το μοντέλο που θα αποτελέσει το baseline μας στο διακριτό χώρο. Πιο συγκεκριμένα, οι τεχνικές με τις οποίες πειραματιζόμαστε και τις οποίες θα θεωρήσουμε γνωστές χωρίς να προχωρήσουμε σε περαιτέρω ανάλυση, είναι οι:

  1)Good-Turing discounting
  2)Absolute Discounting

3)Witten-Bell Discounting
4)Kneser-Ney Discounting

$$
\begin{aligned}
ngram - count & - vocab\ Lexicon.file \\
& - read\ train\_3gram \\
& - order\ 3 \\
& - lm\ WittenBellDis\_LM \\
& - wbdiscount1 \\
& - wbdiscount2 \\
& - wbdiscount3
\end{aligned}
$$

Παράμεροι:
-vocab: διαβάζει το λεξιλόγιο από το αρχείο που δίνεται
-read: διαβάζει το αρχείο που περιέχει τα n-gram counts
-order: θέτει το μέγιστο μέγεθος των N-grams
-lm: το αρχείο εξόδου που περιέχει τα γλωσσικό μοντέλο
-στο παράδειγμα που παρατίθεται, έγινε χρήση του Witten-Bell Discounting που παρέχει και το καλύτερο αποτέλεσμα για το μοντέλο

-Έξοδος αρχείου WittenBellDis_LM
$\backslash data\backslash$
$ngram1 = 2699$
$ngram2 = 313402$
$ngram3 = 259618$
$\backslash 1 - grams:$
$-1.368662\ </s>$
$-99\ <s>\ -1.139609$
$-1.63402\ a\ -0.8487697$
$-4.061882\ ability\ -1.154669$
$-3.722588\ able\ -1.715212$
$-2.630995\ about\ -0.7543153$

$-$ 3.713526 *above* $-$ 0.7057907
⋮
$-$ 0.7781513 *i asked for*
$-$ 1.079181 *i asked him*
$-$ 0.7781513 *it asked state*
$-$ 0.7781513 *it asked the*
$-$ 0.1760913 *previously asked the*
⋮

όπου, η αριστερή στήλη του αρχείου είναι η λογαριθμική, με βάση 10, πιθανότητα (log probability) για κάθε unigram, bigram ή trigram ενώ, η δεξιά στήλη είναι το λογαριθμικό, με βάση 10, βάρος (Log of backoff weight) για κάθε unigram ή bigram

- *Τρίτο στάδιο*: Υπολογίζει το test data perplexity σύμφωνα με το γλωσσικό μοντέλο που έχει εκπαιδεύσει.

$$ngram - ppl\ testData.txt$$
$$- order\ 3$$
$$- lm\ WittenBellDis\_LM$$

Παράμεροι:
-ppl: το αρχείο που περιέχει τα test data
-order: θέτει το μέγιστο μέγεθος των N-grams
-lm: το αρχείο που περιέχει τα γλωσσικό μοντέλο

-Έξοδος αρχείου που περιέχει τα τελικά αποτελέσματα του μοντέλου

$file\ WS94\_6K\_test\_final$:
6000 *sentences*,
152570 *words*,
20420 $OOVs$,

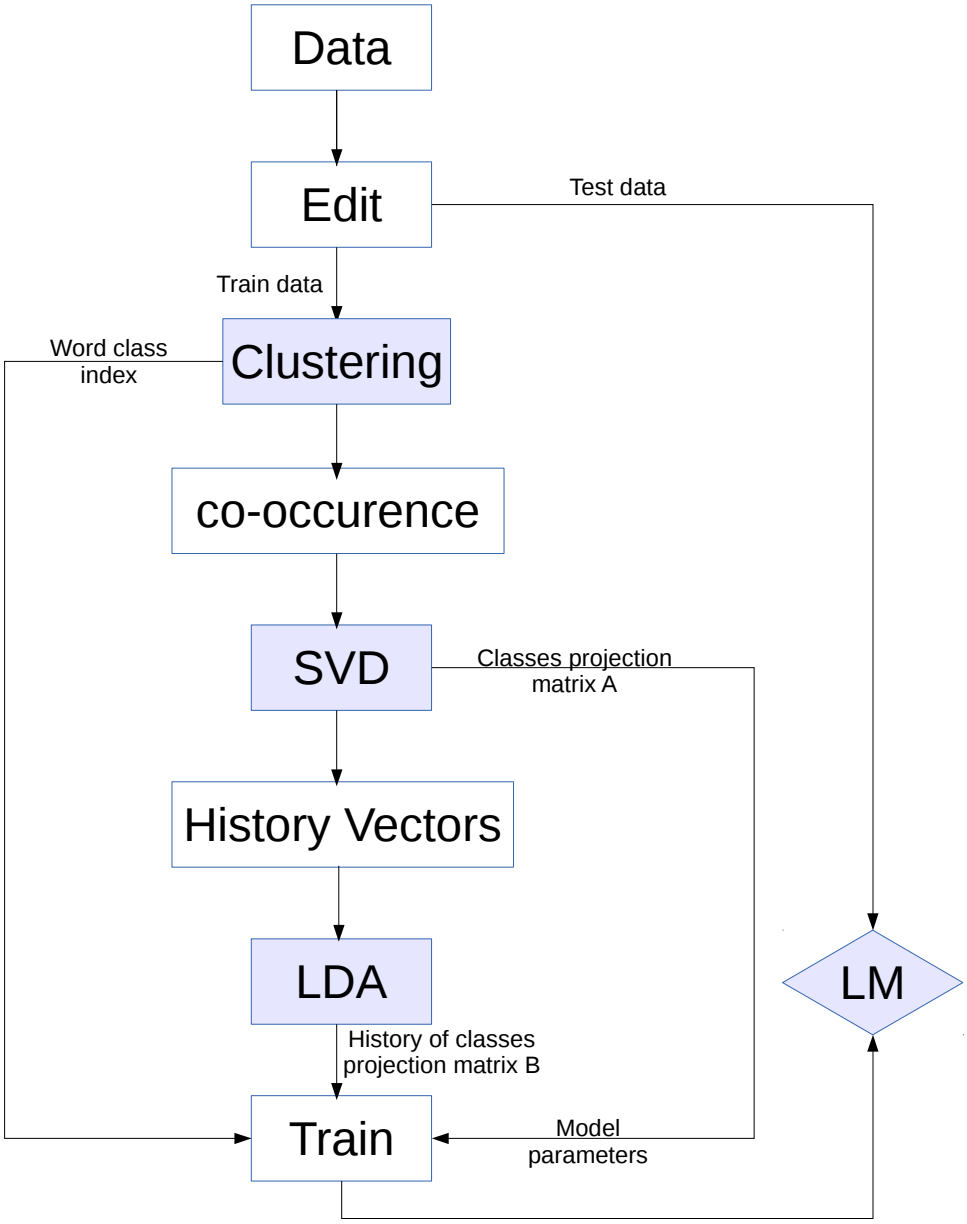
0 *zeroprobs*,
$logprob = -255189$,
$ppl = 70.3374$,
$ppl1 = 85.3206$
όπου $ppl = 10^{\frac{logprob}{sentences+words}}$ και $ppl1 = 10^{\frac{logprob}{words}}$

Discount algorithm	Witten-Bell	without Smoothing
ppl	70.3374	70.9288

Πίνακας 5.2: SRILM perplexity results

5.3 Εκπαίδευση μοντέλου με χρήση HTK

Όπως έχουμε ήδη καλύψει και εξηγήσει και παραπάνω, για όλα τα διαφορετικά μοντέλα, η εκπαίδευση κάθε μοντέλου παραμένει ίδια. Αρχικά, επεξεργαζόμαστε τα δεδομένα μας και συγχρόνως τα διαχωρίζουμε σε δεδομένα εκπαίδευσης και δοκιμής. Εφαρμόζουμε πληθώρα διαφορετικών τεχνικών word clustering πάνω στα δεδομένα εκπαίδευσης και σε επιλεγμένες περιπτώσεις στα δεδομένα δοκιμής όταν είναι απαραίτητο. Η επιλογή του αποδοτικότερου αριθμού κλάσεων αποτελεί τη πρώτη παράμετρο του μοντέλου μας. Στη συνέχεια, κατασκευάζουμε τον class co-occurence matrix που βασίζεται στα bigram που σχηματίζονται μεταξύ δυο διαδοχικών λέξεων. Ο πίνακας αυτός, συμπληρώνεται με βάση την κλάση στην οποία ανήκουν οι λέξεις. Μετέπειτα, εφαρμόζουμε τη τεχνική μείωσης διαστάσεων Singular Value Decompotision για τις M, μεγαλύτερες singular values. Έτσι, δημιουργείται η δεύτερη παράμετρο του μοντέλου μας, τα class vectors. Χρησιμοποιώντας το class mapping που εφαρμόστηκε μέσω του SVD, συλλέγουμε τα διανύσματα ιστορικών για κάθε κλάση τα οποία προβάλουμε σε μια χαμηλότερη διάσταση χρησιμοποιώντας την τεχνική Linear Discriminant Analysis. Ο πίνακας προβολής B, χρησιμοποιείται για την προβολή κάθε ιστορικού κλάσης στο νέο συνεχές χώρο χαμηλότερης διάστασης και είναι η τρίτη παράμετρος του μοντέλου. Τα διανύσματα αυτά που δημιουργούνται, αποτελούν τα δεδομένα εκπαίδευσης, τα οποία χρησιμοποιεί το εργαλείο HTK, το οποίο θα περιγράψουμε σε αυτή την ενότητα, για να εκπαιδεύσει τα mixture models (σχήμα 5.3).



Σχήμα 5.3: Model algorithm

Το Hidden Markov Model Toolkit (HTK) είναι ένα toolkit για τη δημιουργία και το χειρισμό hidden Markov models. Το HTK χρησιμοποιείται κυρίως για έρευνα στην αναγνώριση ομιλίας, παρόλο που έχει χρησιμοποιηθεί για πολλές άλλες εφαρμογές, συμπεριλαμβανομένης της έρευνας σε σύνθεση ομιλίας, αναγνώριση χαρακτήρων και της αλληλούχισης του DNA. Το HTK χρησιμοποιείται σε εκατοντάδες ιστοσελίδες σε όλο τον κόσμο.

Το HTK αποτελείται από ένα σύνολο library modules (βιβλιοθήκες) και εργαλεία που είναι διαθέσιμα σε μορφή κώδικα C. Τα εργαλεία αυτά παρέχουν εξελιγμένες λειτουργίες για την ανάλυση της ομιλίας, HMM training, testing και ανάλυση αποτελεσμάτων. Το λογισμικό που υποστηρίζει τα HMMs χρησιμοποιεί τόσο συνεχείς density mixture Gaussians όσο και διακριτές κατανομές και μπορεί να χρησιμοποιηθεί για την κατασκευή πολύπλοκων συστημάτων HMM. Το HTK περιέχει εκτενή τεκμηρίωση και παραδείγματα. Για οποιαδήποτε πληροφορία, σχετικά με τις λειτουργίες του HTK καθώς και αναλυτική περιγραφή υποστηρίζεται από διαθέσιμη βιβλιογραφία [16].

Το κομμάτι του HTK το οποίο εκμεταλλευόμαστε στη δουλειά μας, αφορά το HMM training χρησιμοποιώντας το λογισμικό που υποστηρίζουν τα HMMs στις συνεχείς density mixture Gaussians με βάση τις προϋποθέσεις που περιγράψαμε στο κεφάλαιο 3.4.

Μετά την εκτίμηση των παραμέτρων του μοντέλου, δηλαδή τον αριθμό κλάσεων και τους πίνακες που προκύπτουν από τις μεθόδους SVD και LDA, εκπαιδεύουμε το μοντέλο μας χρησιμοποιώντας το εργαλείο HTK. Πρέπει να αποφανθούμε στο τρόπο που θα εκπαιδεύσουμε τα μοντέλα μας. Υπάρχουν δύο τρόποι. Αρχικά, εκπαιδεύουμε μια πολυδιάστατη κατανομή Gaussian για κάθε κλάση. Στη συνέχεια, χρησιμοποιούμε το εργαλείο HHEd του HTK για να χωρίσουμε την Gaussian κατανομή σε περισσότερα components. Ο δεύτερος τρόπος αφορά την εκπαίδευση των GMMs class με άμεση χρήση του αριθμού των components, χωρίς να χρησιμοποιούμε τη διαδικασίας διαίρεσης.

Σχετικά με το πρόβλημα που αφορά την υπερπροσαρμογή των δεδομένων (data overfitting) θέτουμε ένα σταθερό κάτω όριο διασποράς (stable variance floor), v=0,1. Έχουμε ήδη αναφερθεί σε αυτό το πρόβλημα, σαν υπολογιστικό πρόβλημα, που εμφανίζεται όταν τα δεδομένα και οι Gaussians έχουν εξαιρετικά μικρές τιμές διασποράς (variance). Η τιμή του floor variance είναι μεγάλη, έτσι υπάρχουν πολλές αποκλίσεις που είναι floored. Αυτό είχε σαν αποτέλεσμα ένα μικρό κομμάτι των πειραμάτων να μην ολοκληρωθεί λόγο αδυναμίας ολοκλήρωσης της εκπαίδευσης από το HTK. Πιο συγκεκριμένα, το εργαλείο HREst τερμάτιζε την εκπαίδευση θεωρώντας ότι τα δεδομένα εκπαίδευσης δεν ήταν κατάλληλα για χρήση (No Usable Training Examples). Σύμφωνα με την ακριβή αναφορά του HTK Book, το συγκεκριμένο λάθος (ERROR [+2226]) εμφανιζόταν καθώς: «κανένα από τα παρεχόμενα δεδομένα εκπαίδευσης δεν θα μπορούσε να χρησιμοποιηθεί για την εκ νέου εκτίμηση του μοντέλου. Τα δεδομένα είναι κατεστραμμένα ή είναι floored». Τα αποτυχημένα αυτά πειράματα εμφανίστηκαν σε ακραία επιλεγμένες τιμές των παραμέτρων ή σε περιπτώσεις κακής ομαδοποίησης των δεδομένων. Έτσι, συνεχίσαμε κανονικά τα πειράματα με διαφορετικές παραμέτρους συμβολίζοντας τα σφάλματα αυτά, στους πίνακες τον αποτελεσμάτων, με το

χαρακτηριστικό H.E.

Τέλος, κάνουμε χρήση του εργαλείου Hinit για να αρχικοποιήσουμε τα GMLM μοντέλα και στη συνέχεια εκτιμούμε τις παραμέτρους του μοντέλου χρησιμοποιώντας το HREst εργαλείο. Χρησιμοποιήσαμε K = {32,64,128} components για την εκτέλεση διαφορετικών μοντέλων.

5.4 Βελτιστοποίηση

Στο κομμάτι της βελτιστοποίησης, αν και αναλογεί ένα πολύ μικρό ποσοστό της ανάλυσης, αποτελεί ένα από τα σημαντικότερα σημεία της υλοποίησης. Χωρίς την προσθήκη αυτής της διαδικασίας δεν θα υπήρχε η δυνατότητα να έχουμε διαθέσιμα πειράματα προς ανάλυση, συγκρίσεις και «προβληματισμούς». Με την έννοια βελτιστοποίηση, αναφερόμαστε κατά κύριο λόγο σε επανεγγραφή του κώδικα ή κομμάτια κώδικα με σκοπό την μείωση της πολυπλοκότητας και κατ΄ επέκταση την ελαχιστοποίηση του χρόνου εκτέλεσης.

Όπως έχουμε ήδη αναφέρει, το βασικό baseline της διπλωματικής μας προέρχεται μια υπάρχουσα δουλειά [3] η οποία κατασκευάζει γλωσσικά μοντέλα στο συνεχή χρόνο, λαμβάνοντας όμως υπόψη κάθε λέξη ξεχωριστά και θεωρώντας ως λεξιλόγιο τις 2700 πιο συχνές λέξεις, αντικαθιστώντας τις υπόλοιπες με την κλάση unk, σε αντίθεση με τη δική μας υλοποίηση όπου λαμβάνουμε υπόψη κλάσεις λέξεων και εργαστήκαμε τόσο για λεξιλόγιο 2700 συχνότερων λέξεων όσο και με χρήση του συνόλου των λέξεων που εμφανίζονταν στα δεδομένα εκπαίδευσης. Όπως λογικά προκύπτει, μεταξύ των δυο εργασιών υπήρχαν αρκετά κομμάτια κώδικα τα οποία παρουσίαζαν αρκετές ομοιότητες. Η αρχική σκέψη ήταν να επαναχρησιμοποιήσουμε αυτά τα κοινά κομμάτια κώδικα με σκοπό να κερδίσουμε χρόνο για την υλοποίηση μας στα υπόλοιπα κομμάτια. Η σκέψη αυτή αποδείχτηκε αδύνατη λόγο της υψηλής πολυπλοκότητας του κώδικα, της μεγάλης διάστασης των διανυσμάτων και κυρίως λόγο της γλώσσας προγραμματισμού που είχε χρησιμοποιηθεί.

Το σημαντικότερο πρόβλημα το οποίο αποτελούσε τροχοπέδη στην εκτέλεση των πειραμάτων της διπλωματικής ήταν ο υπολογισμός που αφορούσε το perplexity στα test data. Ειδικότερα, το πρόβλημα δημιουργούνταν στον τύπο του Bayes που χρησιμοποιείται για την εκτίμηση της trigram log-probability και συγκεκριμένα του παρονομαστή. Παρατηρώντας τη σχέση (5.4), διαπιστώνουμε ότι έχουμε ένα άθροισμα πάνω σε ολόκληρο το λεξιλόγιο (2.700 ή 57.788 λέξεις αντίστοιχα) για κάθε λέξη των test data (164.570). Επομένως, για τον συνολικό υπολογισμό του perplexity απαιτούνταν for-loop 444.339.000 ή 9.510.171.160 επαναλήψεων μόνο για τον παρανομαστή.

Η διαδικασία της βελτιστοποίησης θα μπορούσε να χωριστεί χρονικά σε δυο κομμάτια,

που είναι τα εξής:

$1^{η}$ *βελτιστοποίηση:*

Ο χρόνος εκτέλεσης γι΄ αυτό το κομμάτι υλοποίησης, απαιτούσε 10-12 μέρες (μιλώντας πάντα για λεξιλόγιο 2700 λέξεων) ανάλογα την υπολογιστική ισχύ που ήταν διαθέσιμη. Γνωρίζοντας ότι στη δική μας δουλεία, θα έπρεπε να εκτελέσουμε ένα μεγάλο αριθμό πειραμάτων για διαφορετικές τεχνικές και μεγέθη clustering και για διαφορετικές τιμές στις παραμέτρους SVD και LDA ήταν επιτακτική ανάγκη η βελτιστοποίηση και επανεγγραφή του κώδικα υλοποίησης. Έτσι, σε πρώτη φάση αποφασίσαμε να γράψουμε αυτό το κομμάτι κώδικα (matlab), χρησιμοποιώντας μια ισχυρή script γλώσσα προγραμματισμού (Python) που θα μπορούσαμε να εκμεταλλευτούμε τις αποδοτικές δομές hash table. Στα υπόλοιπα κομμάτια κώδικα, της εργασίας, χρησιμοποιήθηκε επίσης η ίδια γλώσσα προγραμματισμού (Python), έχοντας παρόμοιους χρόνους εκτέλεσης με τα ήδη υλοποιημένα (είχαν υλοποιηθεί σε Perl και Matlab.

Αν και η βελτίωση ήταν αισθητή, οι χρόνοι εκτέλεσης στην περίπτωση χρήσης, ως λεξιλόγιο, του σύνολο των διαφορετικών λέξεων η διάρκεια εκτέλεσης του κώδικα υλοποίησης ήταν τεράστια. Συνοψίζοντας, γι΄ αυτήν την πρώτη προσπάθεια βελτιστοποίησης, τα αποτελέσματα παρουσιάζονται στους πίνακες 5.3 και 5.4.

$2^{η}$ *βελτιστοποίηση:*

Έχοντας ολοκληρώσει το πρώτο στάδιο της βελτιστοποίησης, παρατηρήσαμε ότι το πρόβλημα του μεγάλου χρόνου εκτέλεσης συνεχίζεται να υφίσταται στο κώδικα. Πιο συγκεκριμένα, η καθυστέρηση προέκυπτε και πάλι από τον υπολογισμό του παρανομαστή στο κανόνα του Bayes. Αυτό έχει σαν αποτέλεσμα το πρόβλημα να διογκώνεται πολύ περισσότερο στη χρήση του συνόλου των λέξεων των δεδομένων. Έτσι, αποφασίσαμε την υλοποίηση της συγκεκριμένης εκτίμησης (τον παρονομαστή της σχέσης 5.4) σε διαφορετική γλώσσα προγραμματισμού που θα μας έδινε ακόμα καλύτερους χρόνους εκτέλεσης (χρήση γλώσσας προγραμματισμού C). Τα αποτελέσματα βελτιώθηκαν εντυπωσιακά, όπως φαίνεται και από τους χρόνους που παρουσιάζονται στους πίνακες 5.3 και 5.4.

Πίνακας 5.3: Χρόνοι εκτέλεσης για λεξιλόγιο 2700 λέξεων.

Γλώσσα Προγ.	Χρόνος εκτέλεσης
matlab	10-12 μέρες
Python	2,5-3 μέρες
Python, C	106,8 λεπτά

Πίνακας 5.4: Χρόνοι εκτέλεσης για λεξιλόγιο 57788 λέξεων.

Γλώσσα Προγ.	Χρόνος εκτέλεσης
matlab	-
Python	$>$ 100 μέρες
Python, C	96.99 ώρες

5.5 Αποτελέσματα Πειραμάτων

Μετά την εκπαίδευση του μοντέλου, το τελευταίο στάδιο αφορά τον υπολογισμό του test data perplexity για κάθε γλωσσικό μοντέλο.

$$logP(test\_data) = log[P(S_1) \cdot P(S_2) \cdots P(S_T)] \tag{5.1}$$

Για κάθε test data sentence, υπολογίζουμε την log-probability ως εξής:

$$logP(S_k) = log[P(c(w_1)|c(< s >)c(< s >)) \cdots P(c(w_n)|c(w_{n-2})c(w_{n-1}))] \tag{5.2}$$

$$= log[P(c(w_1)|c(< s >)c(< s >)] + \cdots + log[P(c(w_n)|c(w_{n-2})c(w_{n-1}))] \tag{5.3}$$

Χρησιμοποιώντας του κανόνα του Bayes, υπολογίζουμε την log-probability για κάθε trigram ως εξής:

$$log[P(c(w_k)|c(w_{k-2})c(w_{k-1}))] = log\frac{P(c(w_k))P(c(w_{k-2})c(w_{k-1})|c(w_k))}{\sum_V P(c(v))P(c(w_{k-2})c(w_{k-1})|c)} \tag{5.4}$$

Έτσι, κάνοντας χρήση των σχέσεων 5.1, 5.3 και 5.4 μπορούμε πλέον να υπολογίσουμε το test data perplexity από το τύπο :

$$PPL = e^{\frac{-logprob}{T+W-OOVs}}$$

όπου T είναι ο αριθμός των προτάσεων στα test data, W είναι ο συνολικός αριθμός των λέξεων και OOVs είναι ο αριθμός των λέξεων που δεν βρίσκονται στο λεξιλόγιο (Out-Of-Vocabulary words)

Η έξοδος κάθε πειράματος είναι της μορφής:
ESTIMATING SENTENCES TEST DATA PERPLEXITY
-Sentence No.1 has 12 words with 3 OOVs, logprob = -45.8341166343 and ppl = 97.8476486526 completed in 0.0243251999219 minutes
-Sentence No.2 has 13 words with 4 OOVs, logprob = -48.8370781874 and ppl = 132.119632472 completed in 0.0218238512675 minutes
-Sentence No.3 has 31 words with 2 OOVs, logprob = -152.035337074 and ppl = 158.831607571 completed in 0.0704943656921 minutes
-Sentence No.4 has 34 words with 8 OOVs, logprob = -135.110564375 and ppl = 149.022153614 completed in 0.0554759502411 minutes
⋮
-Sentence No.6000 has 43 words with 7 OOVs, logprob = -219.673650935 and ppl = 378.844450863 completed in 0.0302205840747 minutes

---

-File has 152570 words with 20420 OOVs, nlogprob = -730271.105387 and ppl = 197.566230373 completed in 1.78055226054 hours

Παρατηρώντας το αρχείο εξόδου των πειραμάτων, βλέπουμε ότι κάθε γραμμή περιλαμβάνει τον αριθμό της πρότασης, τον αριθμό των λέξεων της πρότασης (T), μη συμπεριλαμβανομένων των χαρακτήρων <s> και </s>. Επίσης, αναγράφεται ο αριθμός των Out Of Vocabulary-(OOV) λέξεων δηλαδή των λέξεων που είτε έχουν αντικατασταθεί με τη κλάση unk (στο μοντέλο με το μικρό λεξιλόγιο), είτε εμφανίζονται στα test data αλλά όχι στα train data (μοντέλο με το μεγάλο λεξιλόγιο), και αποτελούν τις άγνωστες λέξεις. Οι λέξεις αυτές, που ανήκουν στην κατηγορία OOV δεν λαμβάνουν χώρα στον υπολογισμό της ακρίβειας του μοντέλου. Ο όρος logprob, μας δίνει τη συνολική logprob πιθανότητα της πρότασης, αγνοώντας τις λέξεις της πρότασης που δεν βρίσκονται στο λεξιλόγιο. Στη λογαριθμική πιθανότητα logprob δεν περιλαμβάνεται επίσης οι πιθανότητες για τα tokens

<s> που εισάγονται στην αρχή της υλοποίησης, καθώς τα συγκεκριμένα tokens βρίσκονται στην αρχή κάθε πρότασης με αποτέλεσμα να μην έχουν ιστορικά και να μην μπορούν να εκπαιδευτούν. Έτσι, ο συνολικός αριθμός από tokens γι΄ αυτή τη logprob πιθανότητα στη πρόταση No. 1 προκύπτει από:

NoOfWords - OOVs + sentences = 12 - 3 + 1 = 10, για τη συγκεκριμένη πρόταση. Το perplexity είναι ο γεωμετρικός μέσος όρος της σχέσης 1/probability για κάθε token. Η ακριβής έκφραση είναι:

$$ppl = e^{\frac{-logprob}{(words-OOVs+sentences)}}$$

Αποτελέσματα για λεξιλόγιο 2700 λέξεων

Πριν ξεκινήσουμε την παρουσίαση των αποτελεσμάτων, η ακρίβεια που έχει επιτευχθεί σε προηγούμενες εργασίες (baseline) χωρίς τη χρήση ομαδοποίησης, περιλαμβάνει λεξιλόγιο 2700 συχνότερων λέξεων, επιλογή 100 μεγαλύτερων singular values, προβολή των ιστορικών σε διάσταση 50 και είναι: **ppl=252.1**

Ως σκιασμένα κελιά, αναπαριστώνται οι τιμές οι οποίες βελτιώνουν το baseline μας, ενώ με έντονους χαρακτήρες (bold) η καλύτερη τιμή που έχουμε επιτύχει για κάθε διαφορετική τεχνική ομαδοποίησης.

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίησης SRILM με τις εξής προδιαγραφές (πίνακας 5.5 και εικόνα 5.4):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 64 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.5: Perplexity-SRILM-2700 λέξεις-64G

Classes	1^{st}	2^{nd}	3^{th}	4^{th}
32	285.5(10,5)	267.6(20,10)	240.1(34,20)	H.E.(not SVD,20)
64	270.3(20,10)	H.E.(40,20)	H.E.(66,30)	H.E.(not SVD,40)
128	225.7(40,20)	207.9(60,30)	H.E.(80,40)	H.E.(not SVD,60)
256	215.5(60,30)	204.1(80,40)	204.7(100,50)	205.8(120,60)
512	214.6(80,40)	197.5(100,50)	206.5(120,60)	211.4(140,70)
1024	251.6(80,40)	244.5(100,50)	220.7(120,60)	227.7(140,70)
2048	254.7(80,40)	248.7(100,50)	260.1(120,60)	262(140,70)

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίησης SRILM με τις εξής προδιαγραφές (πίνακας 5.6):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 32 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.6: Perplexity-SRILM-2700 λέξεις-32G

Classes	1^{st}	2^{nd}
128	237.3(40,20)	224.0(60,30)

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίησης SRILM με τις εξής προδιαγραφές (πίνακας 5.7):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 128 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.7: Perplexity-SRILM-2700 λέξεις-128G

Classes	1^{st}
256	211.7(100,50)
512	208.9(100,50)

Σχήμα 5.4: Perplexity-SRILM-2700 λέξεις-64G

Παρατηρώντας τα αποτελέσματα, τα οποία είναι καλύτερα στη περίπτωση χρήσης των 64 components για κάθε Gaussian, επιλέγουμε να δουλέψουμε σε όλα υπόλοιπα πειράματα με χρήση 64 components. Επίσης, διαπιστώνουμε ότι τα καλύτερα αποτελέσματα εμφανίζονται για αριθμό κλάσεων 256 ή 512, ενώ για μικρότερο ή μεγαλύτερο αριθμό τα αποτελέσματα χειροτερεύουν αισθητά.

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίηση SRILM με τις εξής προδιαγραφές (πίνακας 5.8 και σχήμα 5.5):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 64 components
3)λαμβάνουμε υπόψη τις λέξεις με συχνότητα εμφάνισης μεγαλύτερη ή ίση με κάποιο threshold σαν ξεχωριστές κλάσεις και ομαδοποιώντας τις υπόλοιπες λέξεις
4)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension, threshold)

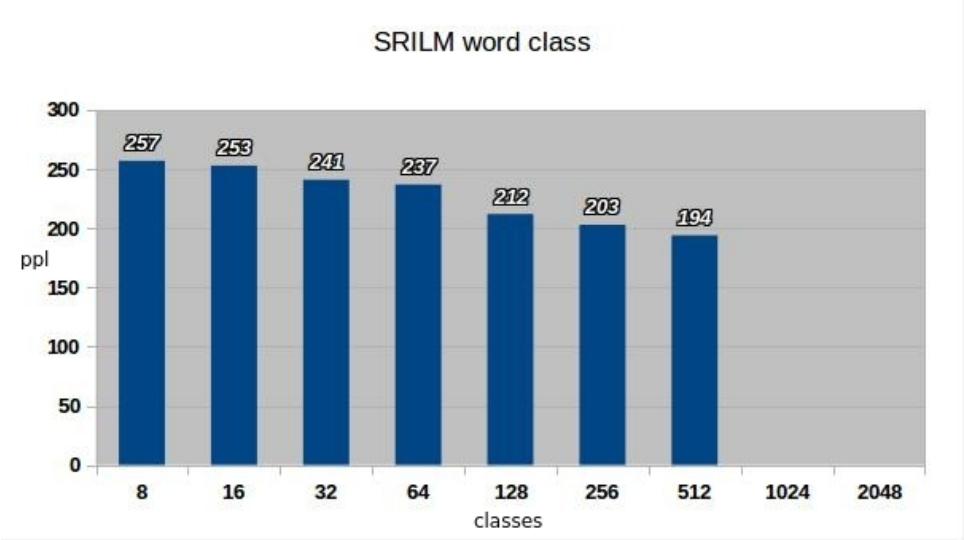
Table 5.8: Perplexity-SRILM-2700 λέξεις με χρήση των συχνότερων λέξεων σαν κλάσεις-64G

Classes	1^{st}	2^{nd}	3^{th}	4^{th}
8	257.7(100,50,700)	267.3(80,40,2000)	279.2(40,20,4000)	292.1(58,30,10000)
16	256.9(100,50,700)	253.8(80,40,2000)	278.6(40,20,4000)	294.3(20,10,10000)
32	248.5(100,50,700)	246(80,40,2000)	241.2(60,30,4000)	265.5(40,20,10000)
64	237.6(1000,50,700)	247.7(100,50,2000)	237.8(80,40,4000)	246.5(60,30,10000)
128	233.1(80,40,700)	230.8(100,50,2000)	212.4(100,50,4000)	220(60,30,10000)
256	230.2(120,60,700)	209.1(100,50,2000)	217.0(100,50,4000)	216.6(100,50,10000)
512	239.2(120,60,700)	212.5(100,50,2000)	211.3(100,50,4000)	216.5(100,50,10000)

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίηση SRILM με τις εξής προδιαγραφές (πίνακας 5.9):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 64 components
3)λαμβάνουμε υπόψη τις λέξεις με συχνότητα εμφάνισης μεγαλύτερη ή ίση με κάποιο threshold σαν ξεχωριστές κλάσεις και ομαδοποιώντας τις υπόλοιπες λέξεις με ταυτόχρονη ομαδοποίηση των λέξεων που ανήκουν στην κλάση unk σε 4 διαφορετικές κλάσεις
4)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension, threshold)
5)εκτέλεση μόνο των καλύτερων αποτελεσμάτων στο προηγούμενο μοντέλο

Table 5.9: Perplexity-SRILM-2700 λέξεις με χρήση των συχνότερων λέξεων σαν κλάσεις και επιπλέον ομαδοποίηση των λέξεων της κλάσης unk-64G

Classes	1^{st}	2^{nd}	3^{th}
128	216.5(100,50,4000)		
256	203.8(100,50,2000)		
512	**194.7(100,50,2000)**	207.1(120,60,2000)	201.5(100,50,4000)



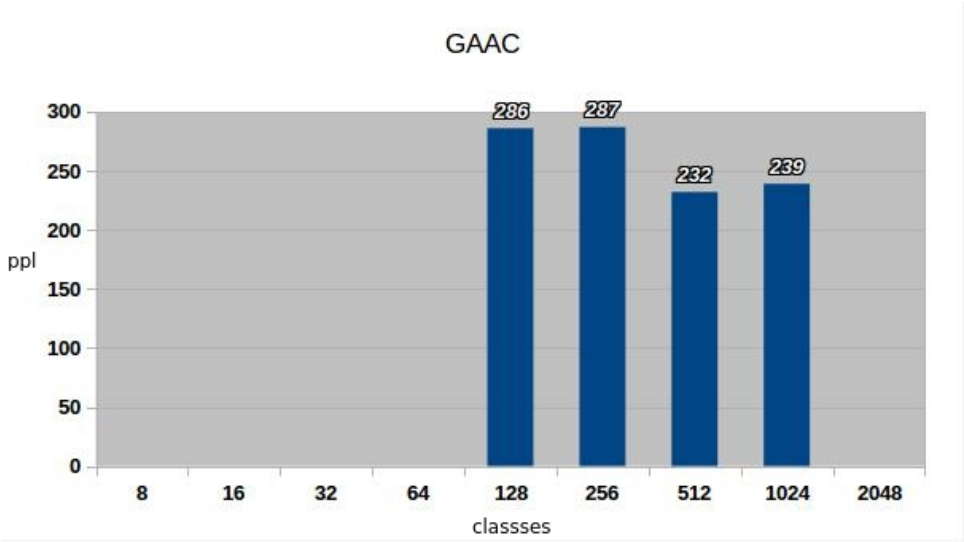
Σχήμα 5.5: Perplexity-SRILM-2700 λέξεις με χρήση των συχνότερων λέξεων σαν κλάσεις-64G

Στο σημείο αυτό διαπιστώνουμε ότι στην εναλλακτική επιλογή των δεδομένων, λαμβάνοντας υπόψη κάθε μια από τις συχνότερες λέξεις (ανάλογα με το threshold) σαν μια ξεχωριστή κλάση, και ομαδοποιώντας τις υπόλοιπες λέξεις τα αποτελέσματα μένουν στα ίδια επίπεδα χωρίς να παρουσιάζουν κάποια βελτίωση. Κάνοντας ένα βήμα παραπάνω, όπου ομαδοποιούμε και τις λέξεις της κλάσης unk σε 4 επιπλέον κλάσεις, τα αποτελέσματα βελτιώνονται περισσότερο από κάθε άλλη περίπτωση. Ειδικότερα, για 512 κλάσεις, SVD dimension=100, LDA dimension=50, και threshold=2000 έχουμε αποτέλεσμα perplexity=194, που αποτελεί το ακριβέστερο μοντέλο. Για threshold=2000, σημαίνει ότι τις λέξεις με αριθμό εμφανίσεων μεγαλύτερο ή ίσο με 2000 λαμβάνονται υπόψη σαν ξεχωριστές κλάσεις.

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίησης GAAC με τις εξής προδιαγραφές (πίνακας 5.10 και σχήμα 5.6):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 64 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.10: Perplexity-GAAC-2700 λέξεις-64G

Classes	1^{st}	2^{nd}	3^{th}	4^{th}
128	303.6(40,20)	286.3(60,30)	294.7(80,40)	-
256	300.3(40,20)	291.5(60,30)	288.3(80,40)	287(100,50)
512	241.3(80,40)	240.6(100,50)	**232.3(120,60)**	240.3(140,70)
1024	253.4(80,40)	258.4(100,50)	242.5(120,60)	239.6(140,70)



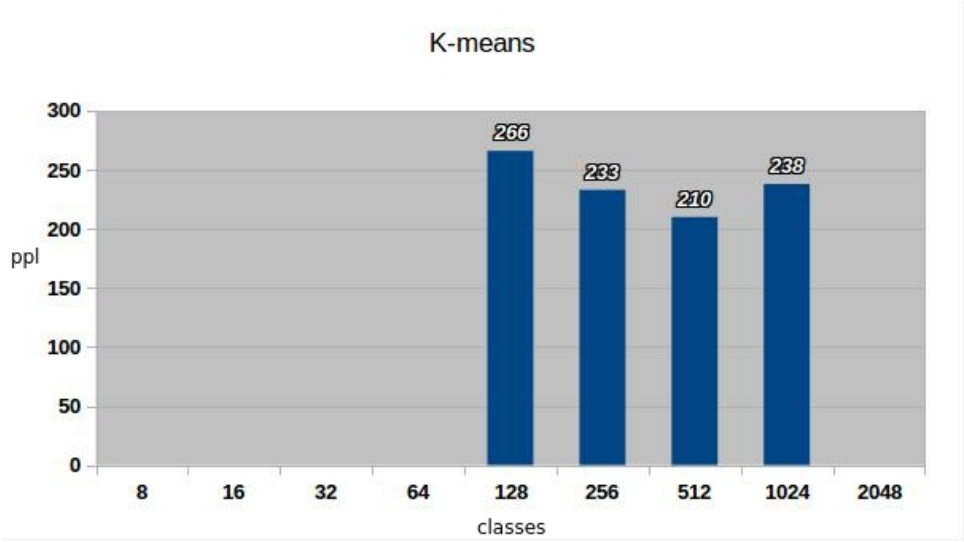
Σχήμα 5.6: Perplexity-GAAC-2700 λέξεις-64G

Παρατηρούμε ότι ο αλγόριθμος ομαδοποίησης GAAG, αν και αυξάνει την ακρίβεια του μοντέλου σε σύγκριση με το baseline, δεν επιφέρει τόσο καλά αποτελέσματα όσο άλλοι αλγόριθμοι που υλοποιήσαμε, καθώς η καλύτερη τιμή που επιτυγχάνει είναι perplexity=232.

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίησης k-means με τις εξής προδιαγραφές (πίνακας 5.11 και σχήμα 5.14):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 64 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.11: Perplexity-K_means-2700 λέξεις-64G

Classes	1^{st}	2^{nd}	3^{th}	4^{th}
128	273.8(40,20)	266.1(60,30)	-	-
256	266.1(40,20)	240.9(60,30)	235.2(80,40)	233.2(100,50)
512	226.2(80,40)	218.3(100,50)	221.3(120,60)	218.8(140,70)
512	216.0(160,80)	213.6(180,90)	**210.1(200,100)**	213.6(220,110)
1024	238.5(80,40)	238.1(100,50)	240.8(120,60)	250.4(140,70)



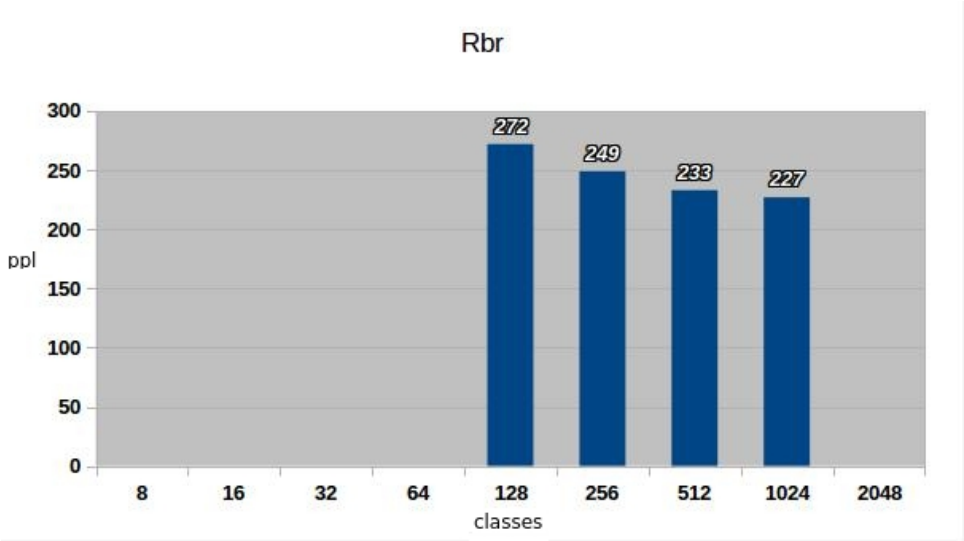
Σχήμα 5.7: Perplexity-K_means-2700 λέξεις-64G

Παρατηρούμε ότι ο αλγόριθμος ομαδοποίησης k-means, αν και αυξάνει την ακρίβεια του μοντέλου σε σύγκριση με το baseline και έχει καλύτερη απόδοση από άλλες τεχνικές που υλοποιήσαμε, δεν επιφέρει καλύτερα αποτελέσματα σε σχέση με την ομαδοποίηση που επιτύχαμε με το SRILM, καθώς η καλύτερη τιμή που επιτυγχάνει είναι perplexity=210.

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίησης rbr με τις εξής προδιαγραφές (πίνακας 5.12 και σχήμα 5.8):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 64 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.12: Perplexity-Rbr-2700 λέξεις-64G

Classes	1^{st}	2^{nd}	3^{th}	4^{th}
128	291.4(40,20)	272.7(60,30)	-	-
256	256.7(60,30)	259.6(80,40)	250.8(100,50)	249.8(120,60)
512	244.4(80,40)	233.7(100,50)	245.9(120,60)	251.9(140,70)
1024	249.8(80,40)	250.5(100,50)	240.5(120,60)	241.1(140,70)
1024	240.9(160,80)	232.5(180,90)	**227.5(200,100)**	230(220,110)



Σχήμα 5.8: Perplexity-Rbr-2700 λέξεις-64G

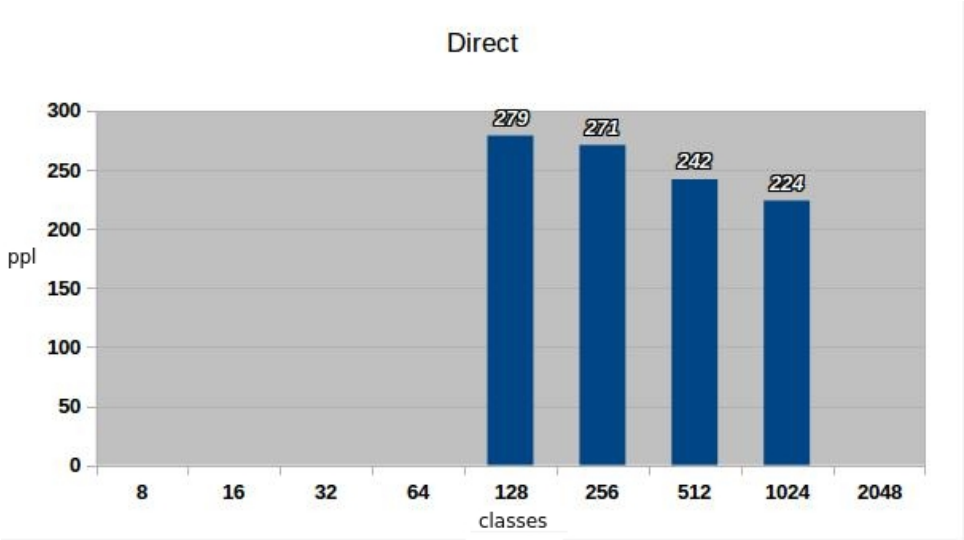
Παρατηρούμε ότι ο αλγόριθμος ομαδοποίησης rbr, αν και αυξάνει την ακρίβεια του μοντέλου σε σύγκριση με το baseline, δεν επιφέρει τόσο καλά αποτελέσματα όσο άλλοι αλγόριθμοι που υλοποιήσαμε, καθώς η καλύτερη τιμή που επιτυγχάνει είναι perplexity=227.

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίησης direct με τις εξής προδιαγραφές (πίνακας 5.13 και σχήμα 5.9):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 64 components

3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.13: Perplexity-Direct-2700 λέξεις-64G

Classes	1^{st}	2^{nd}	3^{th}	4^{th}
128	284.9(40,20)	287(60,30)	279.1(80,40)	-
256	266.1(40,20)	274(60,30)	271.6(80,40)	278.4(100,50)
512	250.1(80,40)	242(100,50)	249.7(120,60)	245.6(140,70)
1024	257.1(100,50)	256.5(120,60)	235.2(140,70)	227.8(160,80)
1024	225.0(180,90)	**224.1(200,100)**	-	-



Σχήμα 5.9: Perplexity-Direct-2700 λέξεις-64G

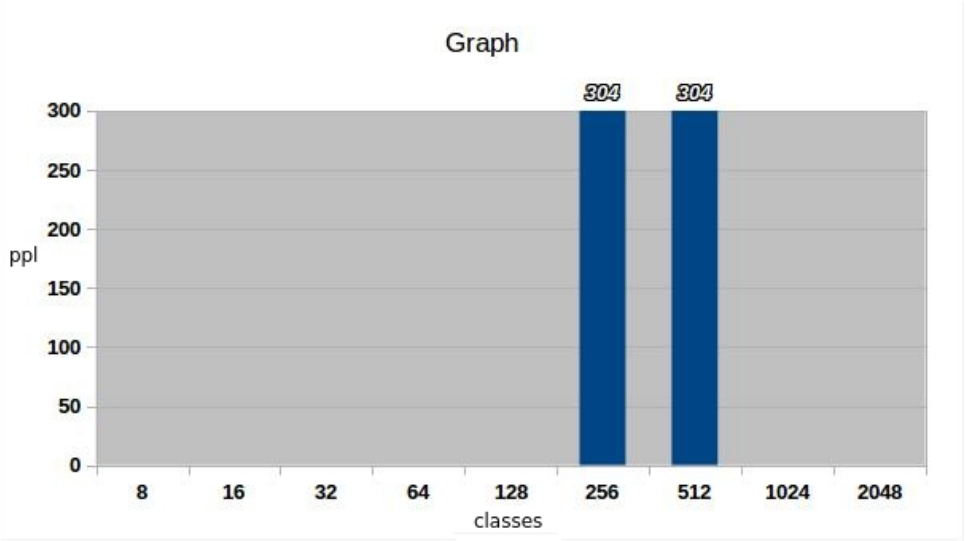
Παρατηρούμε ότι ο αλγόριθμος ομαδοποίησης direct, αν και αυξάνει την ακρίβεια του μοντέλου σε σύγκριση με το baseline, δεν επιφέρει τόσο καλά αποτελέσματα όσο άλλοι αλγόριθμοι που υλοποιήσαμε, καθώς η καλύτερη τιμή που επιτυγχάνει είναι perplexity=224.

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίησης graph με τις εξής προδιαγραφές (πίνακας 5.14 και σχήμα 5.10):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 64 components

3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.14: Perplexity-Graph-2700 λέξεις-64G

Classes	1^{st}	2^{nd}	3^{th}
256	312.6(60,30)	308.9(80,40)	**304.1(100,50)**
512	316.9(80,40)	305.8(100,50)	304.3(120,60)



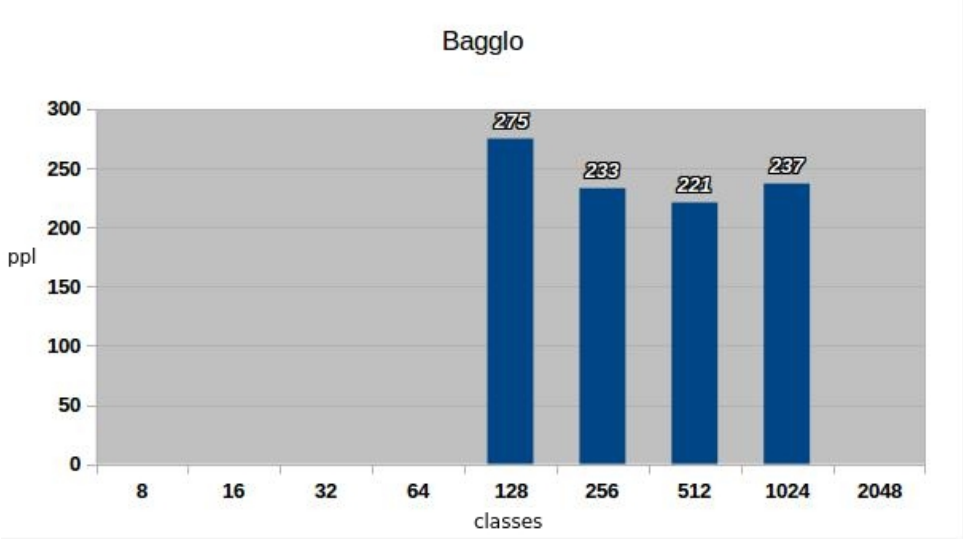
Σχήμα 5.10: Perplexity-Graph-2700 λέξεις-64G

Διαπιστώνουμε ότι ο αλγόριθμος ομαδοποίησης graph, δεν λειτουργεί καθόλου αποδοτικά, καθώς όχι μόνο δεν βελτιώνει τα αποτελέσματα του baseline, αλλά αντιθέτως μειώνει την ακρίβεια επιτυγχάνοντας αποτέλεσμα perplexity=304 .

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίησης bagglo με τις εξής προδιαγραφές (πίνακας 5.15 και σχήμα 5.11):
1)λεξιλόγιο 2700 λέξεων
2)Gaussian με 64 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.15: Perplexity-Bagglo-2700 λέξεις-64G

Classes	1^{st}	2^{nd}	3^{th}
128	287(40,20)	275.8(60,30)	-
256	246.3(80,40)	233(100,50)	234.2(120,60)
512	237.6(80,40)	230.7(100,50)	226.6(120,60)
512	228.3(140,70)	**221.2(160,80)**	225.5 (180,90)
1024	237.1(80,40)	242.1(100,50)	241.7(120,60)



Σχήμα 5.11: Perplexity-Bagglo-2700 λέξεις-64G

Παρατηρούμε ότι ο αλγόριθμος ομαδοποίησης bagglo, αν και αυξάνει την ακρίβεια του μοντέλου σε σύγκριση με το baseline, δεν επιφέρει τόσο καλά αποτελέσματα όσο άλλοι αλγόριθμοι που υλοποιήσαμε, καθώς η καλύτερη τιμή που επιτυγχάνει είναι perplexity=221.

Αποτελέσματα για λεξιλόγιο 57788 λέξεων

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίηση SRILM με τις εξής προδιαγραφές (πίνακας 5.16 και σχημα 5.12):
1)λεξιλόγιο 57788 λέξεων
2)Gaussian με 64 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.16: Perplexity-SRILM-57788 λέξεις-64G

Classes	1^{st}	2^{nd}	3^{th}	4^{th}
32	627.4(10,5)	534.1(20,10)	506.9(33,20)	H.E.(not SVD,20)
64	520.6(20,10)	510.4(40,20)	H.E.(65,32)	H.E.(not SVD,32)
128	417.0(40,20)	389.2(60,30)	384.4(80,40)	404.1(100,50)
256	395.9(60,30)	399.7(80,40)	367.1(100,50)	376.2(120,60)
512	407.5(80,40)	389(100,50)	**364.0(120,60)**	372.7(140,70)
1024	473.8(80,40)	454.4(100,50)	410.2(120,60)	394.1(140,70)
2048	490.2(80,40)	480.9(100,50)	486(120,60)	478.7(140,70)

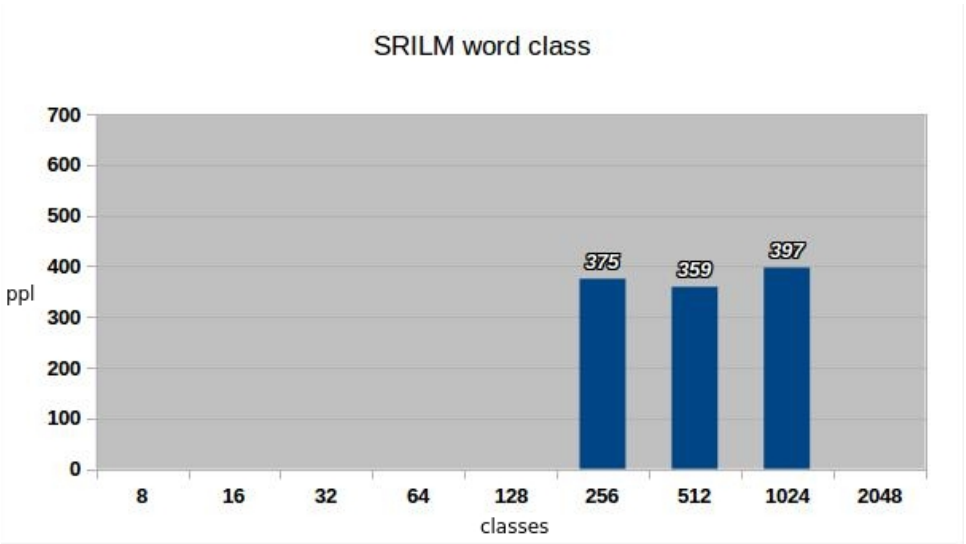
Σχήμα 5.12: Perplexity-SRILM-57788 λέξεις-64G

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίηση SRILM με τις εξής προδιαγραφές (πίνακας 5.17 και σχήμα 5.13):
1)λεξιλόγιο 57788 λέξεων
2)Gaussian με 64 components
3)λαμβάνουμε υπόψη τις λέξεις με συχνότητα εμφάνισης μεγαλύτερη ή ίση με κάποιο threshold σαν ξεχωριστές κλάσεις και ομαδοποιώντας τις υπόλοιπες λέξεις

4)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension,threshold)

Table 5.17: Perplexity-SRILM-57788 λέξεις με χρήση των συχνότερων λέξεων σαν κλάσεις-64G

Classes	1^{st}	2^{nd}	3^{th}	4^{th}
256	401.6(120,60,700)	375.9(100,50,2000)	388.3(100,50,4000)	383.1(100,50,10000)
512	414.6(120,60,700)	378.6(120,60,2000)	365.3(120,60,4000)	**359.4(120,60,10000)**
1024	431.1(140,70,700)	439.7(120,60,2000)	439.2(120,60,4000)	397.4(140,70,10000)



Σχήμα 5.13: Perplexity-SRILM-57788 λέξεις με χρήση των συχνότερων λέξεων σαν κλάσεις-64G

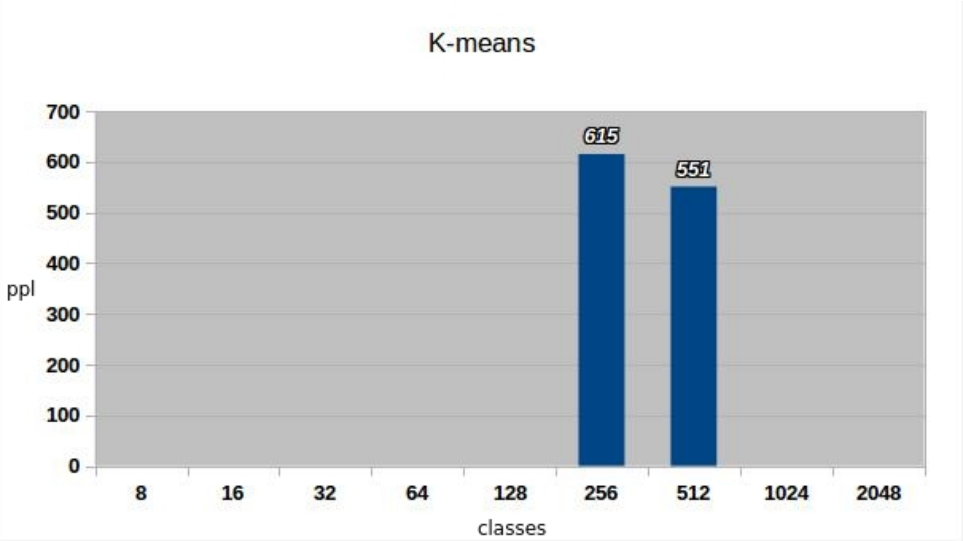
Ο αλγόριθμος ομαδοποίησης GAAC δεν ολοκληρώθηκε επιτυχώς λόγο μεγάλης πολυπλοκότητας του κώδικα με αποτέλεσμα να μην ολοκληρωθεί η εκτέλεσή του.

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίηση k-means με τις εξής προδιαγραφές (πίνακας 5.18 και σχήμα 5.14):
1)λεξιλόγιο 57788 λέξεων
2)Gaussian με 64 components

3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.18: Perplexity-K_means-57788 λέξεις-64G

Classes	1^{st}	2^{nd}	3^{th}
256	615.6(100,50)		
512	556.9(100,50)	572.6(120,60)	**551.4(140,70)**



Σχήμα 5.14: Perplexity-K_means-57788 λέξεις-64G

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίηση rbr με τις εξής προδιαγραφές (πίνακας 5.19 και σχήμα 5.15):
1)λεξιλόγιο 57788 λέξεων
2)Gaussian με 64 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.19: Perplexity-Rbr-57788 λέξεις-64G

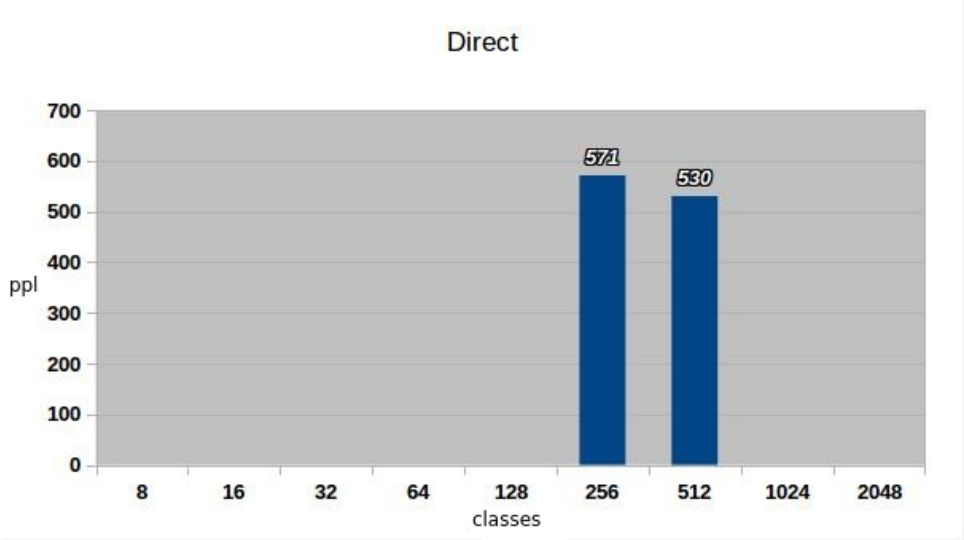
Classes	1^{st}	2^{nd}	3^{th}
256	570.3(100,50)	566.9(120,60)	
512	492.1(120,60)	491.4(140,70)	**489.8(160,80)**

Σχήμα 5.15: Perplexity-Rbr-57788 λέξεις-64G

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίηση direct με τις εξής προδιαγραφές (πίνακας 5.20 και σχήμα 5.16):
1)λεξιλόγιο 57788 λέξεων
2)Gaussian με 64 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.20: Perplexity-Direct-57788 λέξεις-64G

Classes	1^{st}	2^{nd}
256	571.8(100,50)	
512	**530(120,60)**	536.9(140,70)

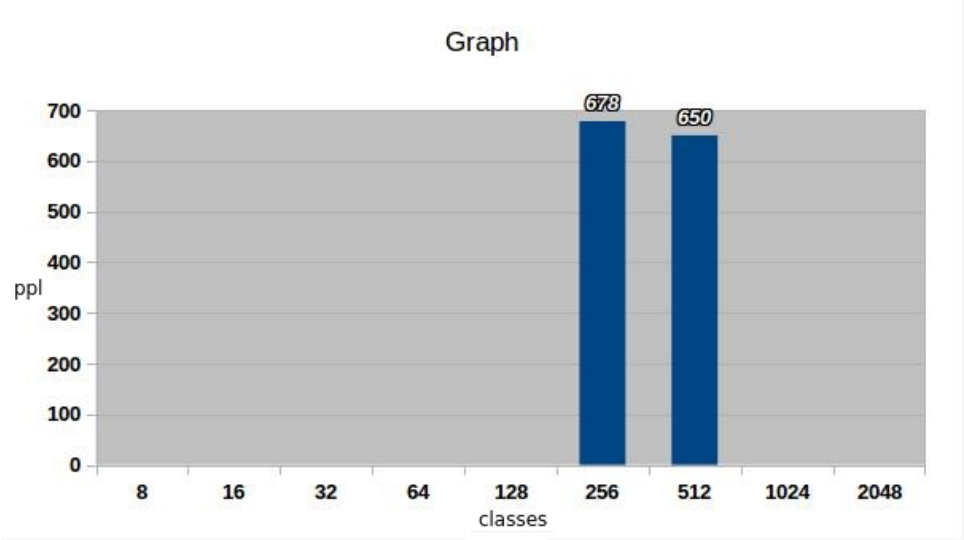


Σχήμα 5.16: Perplexity-Direct-57788 λέξεις-64G

Ακολουθούν τα αποτελέσματα που αφορούν την τεχνική ομαδοποίηση graph με τις εξής προδιαγραφές (πίνακας 5.21 και σχήμα 5.17):
1)λεξιλόγιο 57788 λέξεων
2)Gaussian με 64 components
3)αποτελέσματα της μορφής perplexity(SVD dimension, LDA dimension)

Table 5.21: Perplexity-Graph-57788 λέξεις-64G

Classes	1^{st}
256	678.6(100,50)
512	**650.2(120,60)**



Σχήμα 5.17: Perplexity-Graph-57788 λέξεις-64G

Ο αλγόριθμος ομαδοποίησης bagglo δεν ολοκληρώθηκε επιτυχώς λόγο προβλημάτων μνήμης που εμφάνιζε το εργαλείο Cluto.

Παρατηρώντας τα δεδομένα που προέκυψαν από τη χρήση του μεγάλου λεξιλογίου, διαπιστώνουμε ότι τα αποτελέσματα ήταν αρκετά υποδεέστερα σε σύγκριση με αυτά που επιτεύχθηκαν με το μικρό λεξιλόγιο. Αυτό οφείλεται στο γεγονός ότι στο μοντέλο με το μεγάλο λεξιλόγιο, λαμβάνοντας υπόψη κάθε διαφορετική λέξη που εμφανίζεται στο κείμενο, υπάρχει μεγάλος αριθμός λέξεων με πολύ μικρό αριθμό εμφανίσεων (μια ή δυο εμφανίσεις). Ουσιαστικά, αυτό σημαίνει ότι για τις λέξεις αυτές δεν υπάρχει αρκετή πληροφορία, καθώς λόγο ελάχιστων εμφανίσεων, συνεπάγονται και αντίστοιχα ελάχιστα ιστορικά διανύσματα με αποτέλεσμα να υπάρχει δυσκολία και ανακρίβεια στην ομαδοποίηση και εκπαίδευση των συγκεκριμένων «σπάνιων» δεδομένων.

Πιο συγκεκριμένα, στα αποτελέσματα διαπιστώνουμε ότι η ομαδοποίηση που υλοποιήθηκε με τη χρήση του SRILM επιφέρει και πάλι τα καλύτερα αποτελέσματα επιτυγχάνοντας perplexity=359. Ειδικότερα, το αποτέλεσμα αυτό εκτιμάται στο μοντέλο που θεωρούμε τις συχνότερες λέξεις σαν ξεχωριστές κλάσεις. Στις υπόλοιπες τεχνικές ομαδοποίησης, η ακρίβεια των μοντέλων είναι πολύ χαμηλότερη με «μικρή εξαίρεση» τον αλγόριθμο rbr.

Κεφάλαιο 6

Συμπεράσματα και Προοπτικές

6.1 Ανασκόπηση Διπλωματικής Εργασίας

Σε αυτή την εργασία, αρχικά προτείνουμε γλωσσικά μοντέλα στο συνεχή χώρο στα οποία, τα δεδομένα που αποτελούσαν το λεξιλόγιο ομαδοποιήθηκαν σε κλάσεις. Εφαρμόζουμε διάφορες τεχνικές word clustering (αφού πρώτα έχουμε αντιστοιχίσει τις λέξεις με διανύσματα (word mapping), ενώ σε ορισμένα επιλεγμένα πειράματα επιχειρήσαμε να ομαδοποιήσουμε μόνο ένα μέρος των λέξεων, και τις υπόλοιπες συχνότερες λέξεις να τις χρησιμοποιήσουμε σαν ξεχωριστές κλάσεις. Με βάση τα κύρια μειονεκτήματα των N-grams διακριτών γλωσσικών μοντέλων, την προσαρμοστικότητα και τη γενίκευση, χρησιμοποιούμε τις κατάλληλες μεθόδους mapping για να προβάλουμε τις κλάσεις λέξεων ως μεταβλητές στο συνεχή χώρο. Αρχικά, χρησιμοποιήσαμε ένα μικρό λεξιλόγιο των V = 2700 πιο συχνών λέξεων των δεδομένων εκπαίδευσης (θεωρώντας τις υπόλοιπες λέξεις σαν άγνωστες και τις τοποθετούμε σε μια κλάση unk) όπως επίσης, υλοποιήσαμε την ίδια διαδικασία, χρησιμοποιώντας ως λεξιλόγιο το σύνολό των διαφορετικών λέξεων που εμφανίζονται στα δεδομένα εκπαίδευσης. Επιπλέον, εφαρμόζουμε τεχνικές Singular Values Decomposition και Linear Discriminant Analysis για τη μείωση της διάστασης των χαρακτηριστικών και τον αλγόριθμο Exception Maximazation για την εκπαίδευση των μειγμάτων του μοντέλου. Η εκπαίδευση αυτή, γίνεται με χρήση ενός Gaussian Mixture Model πάνω σε κάθε κλάση. Σε προέκταση του προηγούμενου βήματος, εφαρμόζουμε tying μεθόδους, σε αυτά τα μοντέλα και πειραματιζόμαστε σε διαφορετικά μεγέθη ομαδοποίησης και προβολής των δεδομένων, ώστε να βρούμε τα πιο αποδοτικά για το μοντέλο μας.

Μετά την εκπαίδευση κάθε μοντέλου, εκτιμούμε τη λογαριθμική πιθανότητα για το σύ-

νολο δεδομένων δοκιμής, $logP(testset) = logP(sentence_1) + logP(sentence_2) + \cdots + logP(sentence_n)$, όπου $P(sentence_k) = P(c(w_1)|c(< s >)c(< s >))P(c(w_2)|c(w_1)c(< s >)) \cdots P(c(w_n)|c(w_{n-1})c(w_{n-2}))$. Κάθε μία από τις πιθανότητες των κλάσεων εκτιμάται σύμφωνα με τον κανόνα του Bayes. Τέλος, υπολογίζουμε το perplexity των test data, σύμφωνα με τον τύπο υπολογισμού του SRILM, $PPL(testset) = e^{\frac{-logP(testset)}{(\sharp sentences + \sharp word - \sharp OOVs)}}$

Αξιολογώντας τα αποτελέσματα από τους πίνακες που παραθέσαμε στο προηγούμενο κεφαλαίο, το πρώτο και βασικότερο συμπέρασμα στο οποίο καταλήγουμε είναι ότι η ομαδοποίηση ενισχύει την ακρίβεια του μοντέλου καθώς μειώσαμε την τιμή του perplexity που είχαμε θέσει ως αρχικό μέτρο σύγκρισης ($ppl = 252$). Πιο συγκεκριμένα, για το μικρό λεξιλόγιο, η καλύτερη τιμή ($ppl = 194$) επιτεύχθηκε στην περίπτωση όπου:

- Οι λέξεις με συχνότητα εμφάνισης τουλάχιστον 2000 φορές στο κείμενο (205 τέτοιες λέξεις), τις λαμβάνουμε υπόψη κάθε μια σαν ξεχωριστή κλάση.
- Τις λέξεις που δεν ανήκουν στις 2700 συχνότερες λέξεις, τις τοποθετήσαμε στην κλάση unk.
- Τις λέξεις που δεν ανήκουν σε κάποια από τις δυο προηγούμενες περιπτώσεις, τις ομαδοποιούμε με τη χρήση του εργαλείου SRILM.
- Επιπλέον ομαδοποιούμε, και πάλι με τη χρήση του SRILM, τις λέξεις της κλάσης unk σε 4 επιπλέον κλάσεις.

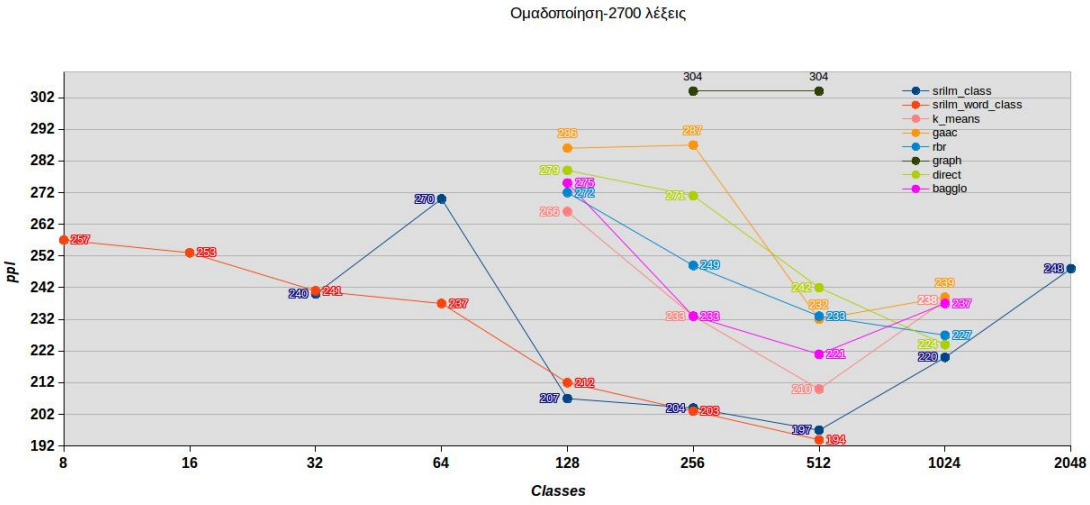
Για το μεγάλο λεξιλόγιο, η καλύτερη τιμή ($ppl = 359$) επιτεύχθηκε στην περίπτωση όπου:

- Οι λέξεις με συχνότητα εμφάνισης τουλάχιστον 10000 φορές στο κείμενο (50 τέτοιες λέξεις), τις λαμβάνουμε υπόψη κάθε μια σαν ξεχωριστή κλάση.
- Τις υπόλοιπες λέξεις, τις ομαδοποιούμε με τη χρήση του εργαλείου SRILM.

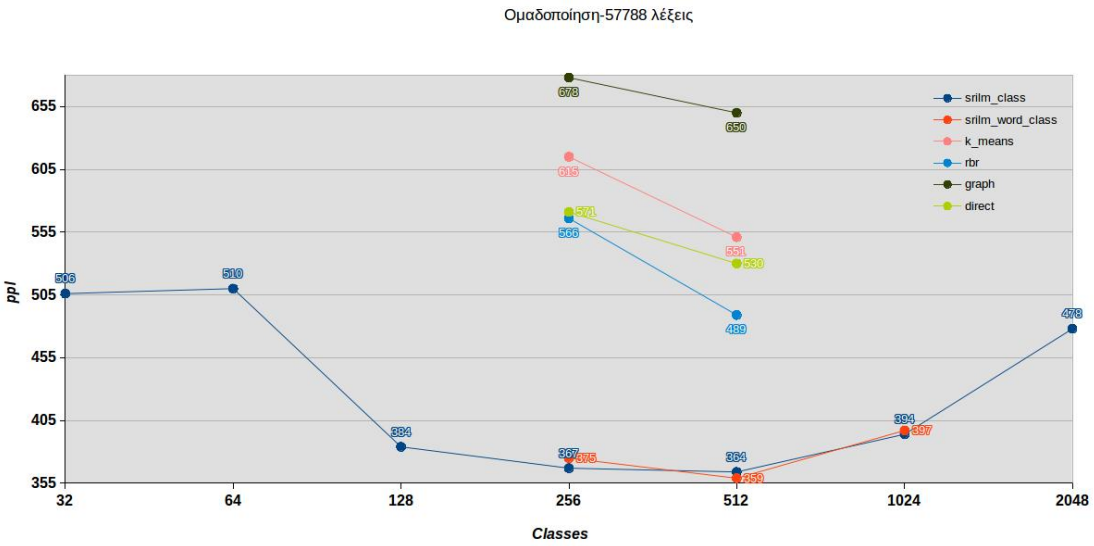
Από την άλλη πλευρά, από τους αλγορίθμους ομαδοποίησης που υλοποιήσαμε, ο μοναδικός που δεν επιτυγχάνει την παραμικρή βελτίωση (τα αποτελέσματα του μοντέλου ήταν πολύ χειρότερα), είναι ο αλγόριθμος Graph/KNN.

Επίσης, παρατηρούμε ότι στις περισσότερες τεχνικές ομαδοποίησης τα καλύτερα αποτελέσματα επιτεύχθηκαν για αριθμό κλάσεων 512, ενώ σε ορισμένες για αριθμό κλάσεων 1024.

Τέλος, διαπιστώνουμε ότι η μοντελοποίηση είναι αρκετά πιο ακριβής στη περίπτωση χρήσης του μικρού λεξιλογίου. Αυτό οφείλεται στο γεγονός ότι στην υλοποίηση του μοντέλου στην οποία χρησιμοποιούμε, το σύνολο των διαφορετικών λέξεων που εμφανίζονται στο κείμενο σαν λεξιλόγιο, υπάρχει μεγάλος αριθμός λέξεων που εμφανίζεται πολύ περιορισμένο αριθμό φορών με αποτέλεσμα να μην υπάρχει ακρίβεια στην ομαδοποίηση αυτών των λέξεων και κατ΄ επέκταση στην εκπαίδευση αυτών των κλάσεων, εμφανίζοντας μεγάλες τιμές στο υπολογισμό του perplexity.



Σχήμα 6.1: Τεχνικές ομαδοποίησης για λεξιλόγιο 2700 λέξεων



Σχήμα 6.2: Τεχνικές ομαδοποίησης για λεξιλόγιο 57788 λέξεων

6.2 Προεκτάσεις για Μελλοντική Έρευνα

Όπως σε κάθε ερευνητική προσπάθεια, έτσι και στη δική μας, δεν υπάρχει τέλος. Υπάρχουν πάντα διάφορες ιδέες και αρκετές παράμετροι που μπορούν να εφαρμοστούν και να δοκιμαστούν αντίστοιχα, με την ελπίδα ότι θα επιφέρουν κάποια βελτίωση στα αποτελέσματα. Η επέκταση της διπλωματικής λοιπόν μπορεί να γίνει σε διάφορες κατευθύνσεις.

Σαν πρώτη πρόταση, που απαιτεί χρόνο και λιγότερη προσπάθεια υλοποίησης, είναι η εκτέλεση περισσότερων πειραμάτων για διαφορετικές παραμέτρους στον αριθμό των κλάσεων, στην επιλογή διαφορετικής τεχνική ομαδοποίησης, στο μέγεθος του μικρού λεξιλογίου (αντί για 2700), στον αριθμό των singular values στη μέθοδο μείωσης διαστάσεων SVD και στο μέγεθος της διάστασης προβολής με την τεχνική LDA.

Μια πολύ ενδιαφέρουσα προέκταση αποτελεί η ταξινόμηση των λέξεων με διαφορετικές κατηγορίες χαρακτηριστικών. Αναλυτικότερα, χρήση του WordNet (μια λεξιλογική βάση για Αγγλικά) με σκοπό την αντιστοίχηση κάθε λέξης με κάποιο διάνυσμα χρησιμοποιώντας μεθόδους hypernym, hyponym, synonym, antonym,.... Άλλη προσέγγιση αποτελεί η ταξινόμηση των λέξεων σύμφωνα με Syntactic Similarity, εκμεταλλευόμενοι το συντακτικό τύπο κάθε λέξης.

Τέλος, θα μπορούσαν να ελεγχθούν εναλλακτικές τεχνικές μείωσης των διαστάσεων οι οποίες θα επέτρεπαν ακόμα καλύτερη εξαγωγή της χρήσιμης πληροφορίας.

Βιβλιογραφία

[1] Afify, M., Siohan, O., Sarikaya, R.: Gaussian mixture language models for speech recognition. IEEE International Conference Acoustics, Speech and Signal Processing (2007) 2, 16

[2] Chen, W.: Building language model on continuous space using gaussian mixture models. unknown (2007) 2, 16

[3] Tsiakas, K.: Language models using continuous distributions. Diploma thesis, Technical University of Crete, Greece (2012) 2, 16, 73

[4] S.Theodoridis, K.Koutroumbas: Pattern Recognition. Academic Press (2009) 13, 14, 48

[5] Federico, M., Cettolo, M.: Efficient handling of n-gram language models for statistical machine translation. In Proc. of the Second Workshop on Statistical Machine Translation (2007) 16

[6] Federico, M., Cettolo, M.: How many bits are needed to store probabilities for phrase-based translation. In Proc. Interspeech (2006) 16

[7] Hsu, B.J.P., Glass, J.: Iterative language model estimation: Efficient data structure & algorithms. In Proc. of the Workshop on Statistical Machine Translation (2008) 16

[8] Hsu, B.J.P., Glass, J.: N-gram weighting: Reducing training data mismatch in cross-domain language model estimation. In Proc. EMNLP (2008) 16

[9] Afify, M., Kingbury, B., Sarikaya, R.: Tied-mixture language modeling in continuous space. Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chap of the Association for Computational Linguistics 16

[10] Schwenk, H., Gauvain, J.L.: Using continuous space language models for conversational speech recognition. ISCA and IEEE Workshop on Spontaneous Speech Processing and Recognition (2003) 16

[11] Brown, P.F., deSouza, P.V., Mercer, R.L., Pietra, V.J.D., Lai, J.C.: Class-based n-gram models of natural language. Computational Linguistics (1992) 42

[12] Ortmanns, P.S., H.Ney, F.Seide, I.Lindam: A comparison of time conditioned and word conditioned search techniques for large vocabulart speech recognition. Philips GmbH Forschungslaboratorien (1992) 42

[13] Cohn, T.: Source code for nltk.cluster.kmeans (2001-2013) http://nltk.org/_modules/nltk/cluster/kmeans.html#KMeansClusterer. 46

[14] Cohn, T.: Source code for nltk.cluster.gaac (2001-2013) http://nltk.org/_modules/nltk/cluster/gaac.html#GAAClusterer. 48

[15] Karypis, G.: CLUTO A Clustering Toolkit. Release 2.1.1 edn. (November 2003) 49

[16] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X.A., Moore, G., Odell, J., et al.: The HTK Book. Cambridge University Engineering Department (2006) 72