Α Ρ Ι Σ Τ Ο Τ Ε Λ Ε Ι Ο Π Α Ν Ε Π Ι Σ Τ Η Μ Ι Ο Θ Ε Σ Σ Α Λ Ο Ν Ι Κ Η Σ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Απόστολος Παπαδόπουλος
Α.Ε.Μ 2916

Εφαρμογή Θεματικής Μοντελοποίησης σε Μαζικά Ανοικτά Διαδικτυακά Μαθήματα

(Applying Topic Modeling to Massive Open Online Courses)

Επιβλέπων:
ΣΤΑΥΡΟΣ ΔΗΜΗΤΡΙΑΔΗΣ
Καθηγητής

ΘΕΣΣΑΛΟΝΙΚΗ, 2021

ΠΕΡΙΛΗΨΗ

Αντικείμενο της παρούσας εργασίας είναι η εφαρμογή αλγορίθμων για μοντελοποίηση θεμάτων στις τελικές απαντήσεις που δίνουν οι μαθητές σε ελληνικά μαζικά ανοικτά διαδικτυακά μαθήματα.

Στα τελευταία χρόνια τα Μαζικά Ανοικτά Διαδικτυακά Μαθήματα έχουν αναπτυχθεί και έχουν λάβει τεράστια δημοσιότητα. Χάρις αυτά τα μαθήματα, εκατομμύρια μαθητές από όλο το κόσμο έχουν πρόσβαση σε εξαιρετικό υλικό. Τις περισσότερες φορές δεν χρειάζεται να πληρώσουμε ή αν χρειάζεται τότε το κόστος είναι πολύ μικρό σε σχέση με άλλες επιλογές εκπαίδευσης. Οι πάροχοι των μαθημάτων, που μπορεί να είναι πανεπιστήμια ή ιδιωτικές πλατφόρμες, έχουν στήσει ένα ολοκληρωμένο πρόγραμμα που απευθύνεται σε χιλιάδες εγγεγραμμένους μαθητές. Από τα πολλά βιβλία και ασκήσεις μέχρι και τα υψηλής ποιότητας βίντεο, ο χρήστης μπορεί να καλύψει τις μαθησιακές του ανάγκες. Επίσης πολλά Massive Open Online Courses (MOOCs) προσφέρουν μαζικές εξετάσεις με τη πιθανότητα λήψεις κάποιας πιστοποίησης και chat rooms που οι μαθητές με ή χωρίς τον καθηγητή, αλληλοβοηθιούνται για να βρουν λύση σε προβλήματα.

Τα MOOCs συνήθως είναι προσβάσιμα όλο το 24ωρο, κάθε μέρα. Το κλίμα συνεργασίας που επικρατεί, προωθεί τους μαθητές να εκφραστούν και να βοηθήσουν ο ένας τον άλλο. Αυτή η συνεχής ανταλλαγή ιδεών δημιουργεί καθημερινά ένα τεράστιο όγκο δεδομένων που συνήθως μένει ανεκμετάλλευτος. Από το γραπτό λόγο το κάθε ατόμου θα μπορούσαμε να δούμε τη θεματική της συνομιλίας και να προβλέψουμε προς τα που πηγαίνει.

Επιπλέον, οι μοντελοποίηση θεμάτων μπορεί να φανεί χρήσιμη στις εξετάσεις. Είναι παράλογο να περιμένουμε μία μικρή ομάδα καθηγητών να εξετάσει τις απαντήσεις κυριολεκτικά χιλιάδων μαθητών. Από την άλλη μεριά ακόμα δεν έχουν αναπτυχθεί αρκετά οι τεχνικές αυτόματης βαθμολόγησης άρα δεν μπορούμε να αφήσουμε το σύστημα να βαθμολογεί όλες τις απαντήσεις. Αυτό που μπορούμε να κάνουμε είναι να χρησιμοποιήσουμε τεχνικές μοντελοποίησης θεμάτων (topic modeling) για να ομαδοποιήσουμε τις απαντήσεις και να προσφέρουμε μια γενική εικόνα στην ομάδα των καθηγητών.

Οι αλγόριθμοι μοντελοποίησης θεμάτων (topic modeling) εκπαιδεύονται χωρίς επίβλεψη σε ένα σύνολο εγγράφων (corpus) που τους δίνουμε. Στη παρούσα εργασία θα ερευνήσουμε πως συμπεριφέρονται δυο γνωστοί αλγόριθμοι topic modeling ο Latent Dirichlet Allocation (LDA) και ο Non-negative Matrix Factorization(NMF ή NNMF). Επίσης θα δοκιμάσουμε τους πιο εξειδικευμένους αλγόριθουμς Gibbs sampling for Dirichlet Mixture Model (GSDMM) και Semantics-assisted Non-negative Matrix Factorization (SeaNMF) που υπόσχονται καλύτερα αποτελέσματα για μικρά κείμενα όπως οι τελικές απαντήσεις των μαθητών.

ABSTRACT

The scope of this paper is to implement algorithms for topic modeling in the final answers given by students in Greek Massively Open Online Courses (MOOC).

In recent years, Mass Online Courses have matured and gained tremendous publicity. Thanks to these courses, millions of students from all over the world have access to exceptional material. Most of the time we do not have to pay or if we need then the cost is very small compared to other educational options. The course providers, which can be universities or private platforms, have set up a curated program aimed at thousands of enrolled students. With a large catalogue of books, exercises and high quality videos, the user can meet his learning needs. Also, many Massive Open Online Courses (MOOCs) offer mass exams with the possibility of obtaining a certification and chat rooms where students, with or without the teacher, help each other to find solutions to problems.

MOOCs are usually accessible 24 hours per day, every day. The prevailing atmosphere of cooperation promotes students to express themselves and help each other. This constant exchange of ideas creates a huge amount of data on a daily basis that is usually left untapped. From the written word of each person we could see the topic of the discussion and predict where it is going.

Moreover topic modeling may be helpful in exams. It is unreasonable to expect a small group of teachers to evaluate the answers of literally thousands of students. On the other hand, the automatic scoring techniques have not been developed enough therefore we cannot let the system score all the answers. What we can do is use topic modeling techniques to group the answers and give an overview to the teaching team.

Topic modeling algorithms are trained without supervision in a set of documents (corpus) that we give them. In this paper we will investigate how two well-known topic modeling algorithms behave: Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF or NNMF). We will also test more specialized such as Gibbs sampling algorithms for Dirichlet Mixture Model (GSDMM) and Semantics-assisted Non-negative Matrix Factorization (SeaNMF) which promise better results for short texts such as students' final answers.

Ε Υ Χ Α Ρ Ι Σ Τ Ι Ε Σ

Πριν την παρουσίαση των αποτελεσμάτων της παρούσας εργασίας, αισθάνομαι την υποχρέωση να ευχαριστήσω ορισμένους από τους ανθρώπους που γνώρισα, συνεργάστηκα μαζί τους και έπαιξαν πολύ σημαντικό ρόλο στην πραγματοποίησή της.

Πρώτα από όλα, θέλω να ευχαριστήσω θερμά τον επιβλέποντα Καθηγητή κ. Σταύρο Δημητριάδη για τη καθοδήγηση και την επιείκεια του. Επίσης ευχαριστώ τον Δημήτρη Τζίμα για τη βοήθεια του στο ξεκίνημα της εργασίας και τον Στέργιο Τέγο για τη παροχή των dataset.

Τέλος, ευχαριστώ την οικογένειά μου, όπως επίσης και όλους αυτούς τους κοντινούς μου ανθρώπους που με στήριξαν αυτή τη περίοδο.

12 Μαρτίου, 2021

Απόστολος Παπαδόπουλος

ΠΕΡΙΕΧΟΜΕΝΑ

Λιστα Σχηματων

Λιστα Πινακων

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

ΕΙΣΑΓΩΓΗ

1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΤΗΣ ΠΑΡΟΥΣΑΣ ΕΡΓΑΣΙΑΣ

Το αντικείμενο της παρούσας εργασίας είναι να εφαρμόσουμε αλγόριθμους θεματικής μοντελοποίησης (topic modeling) στις τελικές απαντήσεις μαθητών σε ελληνικά μαζικά ανοικτά διαδικτυακά μαθήματα. Κύριος στόχος της έρευνας μας είναι η εφαρμογή τεσσάρων αλγορίθμων topic modeling, που είναι οι:

- Ο δημοφιλής αλγόριθμος Latent Dirichlet Allocation (LDA), που βασίζεται σε πιθανότητες εμφάνισης των λέξεων.
- Ο αλγόριθμος Non-negative Matrix Factorization(NMF ή NNMF) που χρησιμοποιεί γραμμική άλγεβρα για να παράγει θέματα.
- Οι Gibbs sampling for Dirichlet Mixture Model (GSDMM) και Semantics-assisted Non-negative Matrix Factorization (SeaNMF) που είναι εξειδικευμένες εκδοχές του LDA και NMF αντίστοιχα και υπόσχονται καλά αποτελέσματα για μικρά κείμενα όπως οι τελικές απαντήσεις των μαθητών.

Επιπλέον:

1. Θα χρησιμοποιήσουμε στατιστική ανάλυση κειμένου για να παράγουμε κάποια αποτελέσματα για τις συνομιλίες των μαθητών κατά τη πρόοδος τους προς τη λύση.
2. Θα ξεκινήσουμε από την αρχή με τα έγγραφα όπως μας έχουν δοθεί και θα τα μετατρέψουμε σε σύνολα δεδομένων ώστε να είμαστε έτοιμοι να εφαρμόσουμε τους αλγόριθμους.
3. Θα συγκρίνουμε τα αποτελέσματα των τεσσάρων τεχνικών.

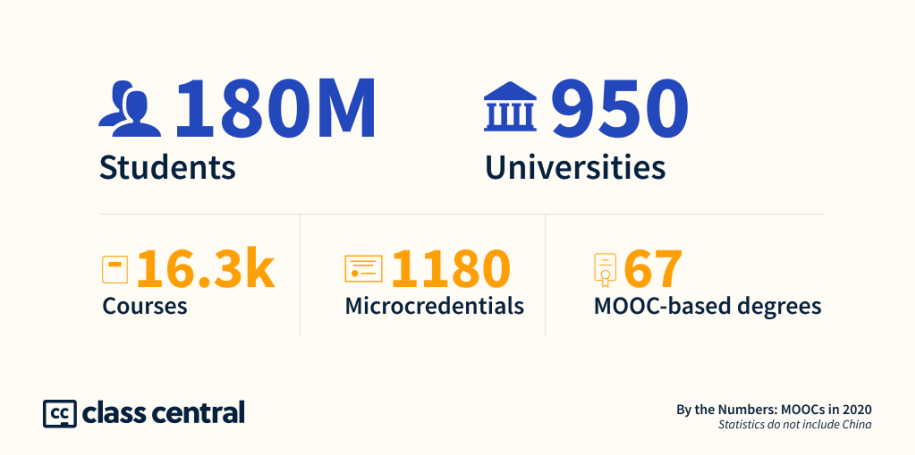
Η εργασία δομείται σε κεφάλαια ως εξής:

- Στο Κεφάλαιο 2 "*Σύνολο Δεδομένων (Dataset)*", παρουσιάζουμε τη δομή των δεδομένων που έχουμε στη κατοχή μας, εξηγούμε τι σημαίνουν οι διάφορες μεταβλητές και εισάγουμε μία πολιτική εκκαθάρισης των μη εγκύρων δεδομένων.
- Στο Κεφάλαιο 3 "Ανάλυση Κειμένου", παρουσιάζουμε τα αποτέλεσμα από τη στατιστική ανάλυση και επισημαίνουμε κάποια συμπεράσματα.
- Στο Κεφάλαιο 4 "Topic Modeling", κάνουμε μία εισαγωγή στη μοντελοποίηση θεμάτων (topic modeling) και εξηγούμε τους τέσσερις αλγόριθμους. Έπειτα ξεκινάμε τη προ-επεξεργασία των τελικών απαντήσεων των μαθητών ώστε να μετατραπούν σε ένα σύνολο δεδομένων έτοιμο για εκτέλεση. Μετά τη παραγωγή των αποτελεσμάτων εξηγούμε για ποιο λόγο είναι δύσκολη η μέτρηση της απόδοσης στο topic modeling.
- Στο Κεφάλαιο 5 "Συμπεράσματα", κάνουμε μια ανακεφαλαίωση των παραπάνω και βγάζουμε τα τελικά μας συμπεράσματα.
- Στο Παράρτημα Ι παρουσιάζονται αλφαβητικά η βιβλιογραφία και οι δικτυακοί τόποι που αναφέρονται στην εργασία.

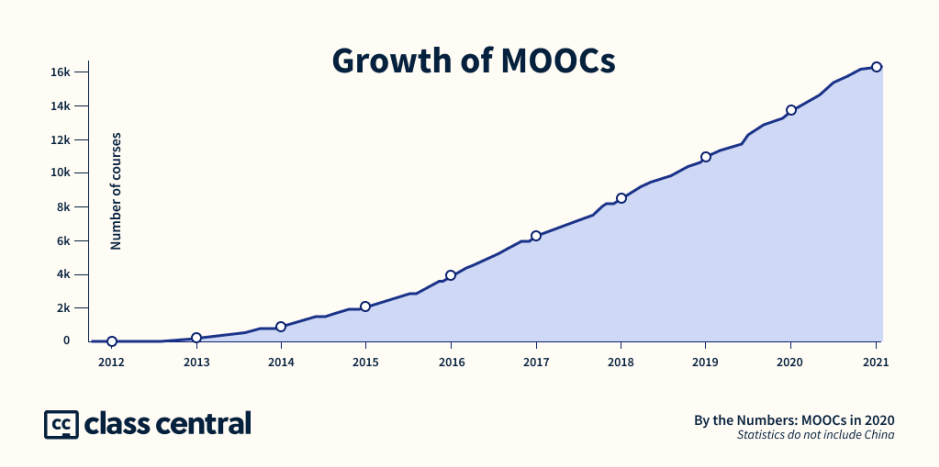
- Στο Παράρτημα II παρουσιάζεται ο δικτυακός τόπος που είναι διαθέσιμος ο κώδικας.
- Στο Παράρτημα III παρουσιάζονται τα ακρωνύμια τα οποία χρησιμοποιούνται σε αυτή την εργασία για την διευκόλυνση του αναγνώστη.
- Στο Παράρτημα IV παρουσιάζεται το γλωσσάριο ξενικών όρων οι οποίοι χρησιμοποιούνται σε αυτή την εργασία για την διευκόλυνση του αναγνώστη.
- Στο Παράρτημα V παρουσιάζεται το ευρετήριο των όρων οι οποίοι χρησιμοποιούνται σε αυτή την εργασία για την διευκόλυνση του αναγνώστη.

1.2 ΕΙΣΑΓΩΓΗ ΣΤΑ MOOC

Το 2020 κατέγραψε νούμερα ρεκόρ για τα Massive Open Online Courses(Μαζικά ανοικτά διαδικτυακά μαθήματα) ή MOOC καθώς έφερε πάνω από 60 εκατομμύρια νέους μαθητές στις δημοφιλέστερες πλατφόρμες. Είναι ένας τομέας που παρουσιάστηκε 2008 από τον Dave Cormier. Μέσα στα χρόνια, οι πλατφόρμες που υποστηρίζουν τέτοιου τύπου μαθήματα έχουν ωριμάσει και λόγου της πανδημίας του 2020 όλο και περισσότεροι καταφθάνουν με σκοπό να μάθουν κάποια τέχνη.



Εικόνα 1: Στατιστικά για MOOC 2020 (Dhawal Shah 2020)



Εικόνα 2: Αύξηση μαθημάτων ανά χρονιά (Dhawal Shah 2020)

Σε αυτά τα μαζικά ανοικτά διαδικτυακά μαθήματα συνήθως μπορεί να μπει ο καθένας και να λάβει υψηλής ποιότητας εκμάθηση. Αρκετά μαθήματα (courses) έχουν δημιουργηθεί από πανεπιστήμια ή από άλλες αξιόπιστες πήγες, σε αυτά ο μαθητής μπορεί να βρει βίντεο, βιβλία, αρχεία και τεστ και ονομάζονται OpenCourseWare (OCW). Ένα βασικό πλεονέκτημα σε σχέση με την αυτομάθηση (self-learning) είναι ότι τα MOOC έχουν επιλεγμένη λίστα κεφαλαίων, συγκεκριμένο πρόγραμμα και περιεχόμενο όπως chat rooms.

Με τα χρόνια έχουν γίνει έρευνες ώστε να δούμε σε τι μπορεί να εξελιχθεί αυτός ο τομέας και αν θα μπορέσει να γίνει νέα πηγή προσβάσιμης εκπαίδευσης. Έχει προταθεί ο όρος MOOC να διαχωριστεί σε:

- cMOOC = Connectivist MOOC, όλοι οι συμμετέχοντες είναι και δάσκαλοι (παρέχουν περιεχόμενο). Στόχος η μάθηση, όχι η πιστοποίηση.

- xMOOC = κλασικό μοντέλο, ο δάσκαλος = video, το περιεχόμενο (OCW) παρέχεται από το δάσκαλο πριν την έναρξη του MOOC.

Τα MOOC διαφέρουν πολύ από online μαθήματα που θα μπορούσαν να γίνουν σε ένα πανεπιστήμιο ή για μία ομάδα ανθρώπων. Όπως λέει και ο τίτλος τα μαθήματα είναι μαζικά, αφορούν εκατοντάδες ή χιλιάδες μαθητές ταυτόχρονα και από διαφορετικά πλαίσια γνώσεων. Πρόκειται για μία ομαδική προσπάθεια που όλοι αλληλοβοηθούνται (connective knowledge). Παράγονται τεράστιοι όγκοι δεδομένων από μηνύματα και απαντήσεις σε τεστ που είναι αδύνατο να διαχειριστούν από ένα μόνο καθηγητή. Τα MOOC χρειάζονται 24x7 υποστήριξη και για αυτό γίνετε χρήση τεχνητής νοημοσύνης (artificial intelligence) υπό τη μορφή chat bot και τεχνικών βαθμολόγησης.

1.3 ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ

Η ανάλυση κειμένου έχει στόχο την εξαγωγή δομημένης πληροφορίας από ένα σύνολο αδόμητων κείμενων. Αυτή η πληροφορία μπορεί ευκολά να χρησιμοποιηθεί αργότερα

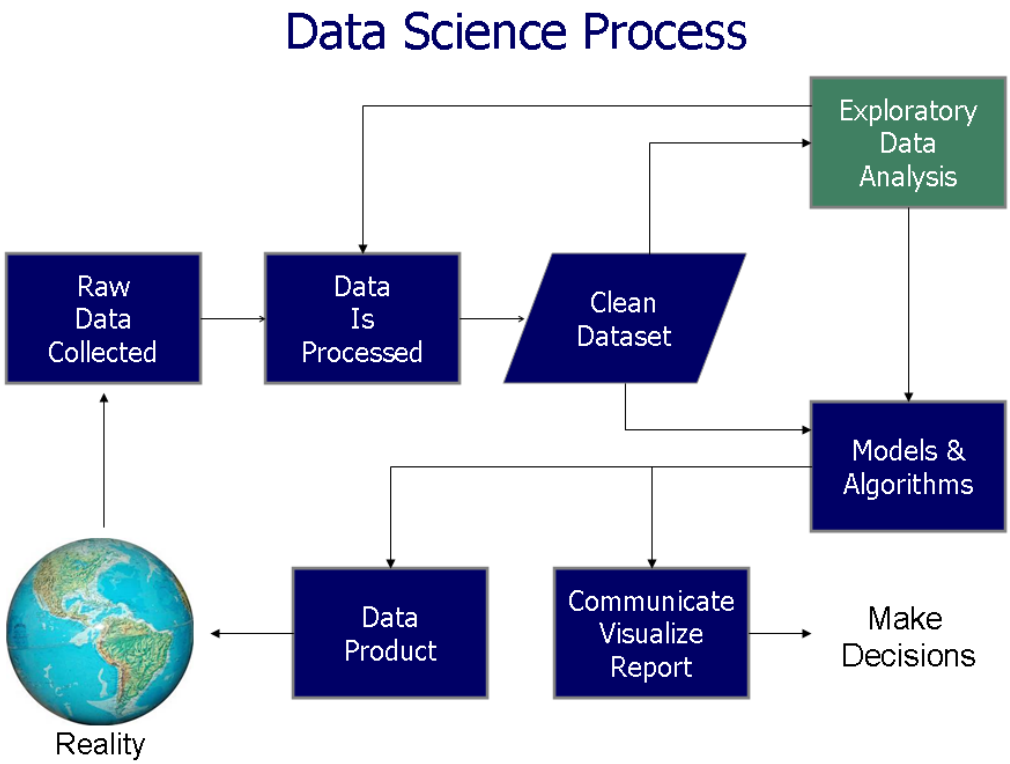
στο να βγάλουμε συμπεράσματα για το κείμενο το οποίο αναλύουμε ή να κάνουμε μια υπόθεση για ένα νέο κείμενο βάση των προηγουμένων.

Η ανάλυση κειμένου θα μπορούσε καλυτέρα να διαχωριστεί σε 2 βασικούς κλάδους:

- Εξόρυξη κειμένου (text mining) : είναι η διαδικασία στην οποία το σύστημα παίρνει ένα σύνολο αδόμητων δεδομένων και προσπαθεί να βρει ομάδες και μοτίβα. Στην εργασία μας θα το χρησιμοποιήσουμε ώστε να βρούμε θεματικές αναμεσά στις τελικές απαντήσεις.
- Αναλυτικά στοιχεία κειμένου (text analytics) : είναι το σύνολο των τεχνικών που χρησιμοποιεί ώστε να βγάλουμε διαφορά στατιστικά αποτέλεσμα. Στην εργασία μας θα τα χρησιμοποιήσουμε για τους στατιστικούς στόχους που έχουν τεθεί ώστε να βγάλουμε συμπεράσματα από τις μεγάλες αριθμητικές αποκλίσεις.

Συνεχίζοντας, όποτε χρησιμοποιούμε τον ορό 'Ανάλυση Κειμένου' θα συμπεριλαμβάνει και τους 2 κλάδους.

1.3.1 Βήματα Ανάλυσης Κειμένου



Εικόνα 3: Διάγραμμα της επιστήμης ανάλυσης δεδομένων από το *Doing Data Science*, από Schutt & O'Neil (2013)

Για να κάνουμε σωστά ανάλυση κειμένου χρειάζεται να περάσουμε από τα εξής βήματα:

1. **Data Gathering:** Συλλέγουμε εάν μεγάλο όγκο εγγράφων (corpus) το οποία θέλουμε να τα αναλύσουμε και να βγάλουμε κάποια συμπεράσματα για αυτά.
2. **Data Preparation:** Πρόκειται για την εκκαθάριση και τη σωστή μορφοποίηση των εγγράφων. Υπάρχει μεγάλη πιθανότητα τα δεδομένα που έχουμε να είναι ανακατεμένα και να έχουμε πληροφορία άχρηστη για την ανάλυση που θα κάνουμε. Επίσης τα δεδομένα μας καλό θα ήταν να τα μορφοποιήσουμε ώστε να μην επηρεάσουν με λάθος τρόπο τα αποτελέσματα μας. Το πετυχαίνουμε με το να βγάλουμε μικρές λέξεις, όλα τα γράμμα να γίνου είτε πεζά είτε κεφάλαια και να βρούμε τη ριζά της λέξης.
3. **Text Analytics:** Τώρα που έχουμε το σύνολο των δεδομένων καθαρισμένο και έτοιμο, μπορούμε να ξεκινήσουμε με την ανάλυση. Οι τεχνικές ανάλυσης που μας ενδιαφέρουν είναι η στατιστική ανάλυση κειμένου και η ομαδοποίηση με machine learning.
4. **Result Visualization:** Παρουσιάζουμε τα αποτελέσματα με τέτοιο τρόπο ώστε να είναι κατανοητό από κάποιον αναλυτή. Δίνουμε ετικέτες σε αριθμητικά δεδομένα και παράγουμε γραφήματα ώστε να μπορούμε να βρούμε συνδέσεις και να καταλάβουμε για ποιο λόγο βγήκαν αυτά τα αποτελέσματα.
5. **Decision Making:** Αφού βρούμε αποτελέσματα που μας βοηθάνε με την ερευνά μας, μπορούμε να παράγουμε αποφάσεις που υποστηρίζονται από πραγματικά δεδομένα.

1.3.2 Χρήση Ανάλυσης Κειμένου στην Εκπαίδευση

- Οι Ye Chen, Bei Yu, Xuewei Zhang, Yihan Yu ερευνούν μια μέθοδο βασισμένη στη Μοντελοποίηση Θεμάτων που στόχο έχει να βρει θεματικές και μοτίβα σε βαθμολογημένα κείμενα που οι μαθητές εξέφραζαν τους προβληματισμούς τους για συγκεκριμένα θέματα. Στο τέλος τις έρευνας αναπτύχθηκε ένα μοντέλο που προσπαθούσε να προβλέψει τη βαθμολογία του κειμένου βάση των θεμάτων που ανέλυε. (Chen, 2016)
- Οι Jovita M. Vytasek, Alyssa F. Wise, Sonya Woloshen προσπαθούν χρησιμοποιώντας topic modeling να βοηθήσουν τους εκπαιδευτικούς να προηγηθούν ευκολότερα στις συζητήσεις που γίνονται στα MOOC. Ο αλγόριθμος κατηγοριοποιούσε τις συζητήσεις έτσι ώστε να ήταν ευκολότερο αργότερα να καταλάβουμε το γενικότερο θέμα της συζητήσεις και να δώσουμε τη κατάλληλη απάντηση. (Vytasek, 2017)
- Οι Philip Resnik, Anderson Garron και Rebecca Resnik χρησιμοποιούν τον αλγόριθμο Latent Dirichlet Allocation (LDA) μαζί με τη βοήθεια του Pennebaker's Linguistic Inquiry and Word Count (LIWC) λεξικού βρίσκουν θεματικές που αφορούν θέματα σχετικά με τη ψυχολογία του φοιτητή σε κλινικές μελέτες. (Resnik, 2013)
- Οι Swapna Gottipati, Venky Shankararaman & Jeff Rongsheng Lin, μας παρουσιάζουν μια ολοκληρωμένη λύση για την εξαγωγή χρήσιμων προτάσεων από μαθητές για τη βελτίωση των μαθημάτων. Κατέληξαν στο συμπέρασμα ότι το δέντρο απόφασης ήταν η καλύτερη λύση. (Gottipati, 2018)

- Οι Nurbiha A Shukor και Zaleha Abdullah ερευνούν πως η χρήση learning analytics μπορεί να βοηθήσει στη καλύτερη δόμηση των MOOC. Βρήκαν ότι ένα από τα σημαντικά μέρη ενός μαθήματος είναι η πρώτη σελίδα διότι μία καλή πρώτη σελίδα θα ενθαρρύνει τον μαθητή να συνεχίσει παρακάτω. Επίσης παρατήρησαν ότι οι δραστηριότητες χεριάζετε να ξεκινάνε με εύκολα πράγραμα και σταδιακά να γίνονται όλο και ποιο πολύπλοκα. (Shukor, 2019)
- Οι Ling Wang, Gongliang Hu and Tiehua Zhou παρουσιάζουν ένα μοντέλο σημασιολογικής ανάλυσης για την παρακολούθηση των συναισθηματικών τάσεων των μαθητών. Μέσω συναισθηματικών ποσοτικοποιήσεων και υπολογισμών μηχανικής μάθησης, η πιθανότητα αποφοίτησης μπορεί να προβλεφθεί για διαφορετικά στάδια μάθησης σε πραγματικό χρόνο. Ειδικά για μαθητές με συναισθηματικές τάσεις, θα μπορούσαν να γίνουν προσαρμοσμένες οδηγίες για τη βελτίωση των ποσοστών ολοκλήρωσης και αποφοίτησης. (Wang, 2018)
- Οι Christopher G. Brinton, Mung Chiang, Shaili Jain, Henry Lam, Zhenming Liu, Felix Ming Fai Wong στοχεύουν στη βελτίωση της ποιότητας της μάθησης μέσω διαδικτυακών φόρουμ συζήτησης, επινοώντας μεθόδους για τη διατήρηση των δραστηριοτήτων του φόρουμ και για τη διευκόλυνση της εξατομικευμένης μάθησης. (Brinton, 2014)

ΚΕΦΑΛΑΙΟ 2: ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ (DATASET)

ΣΥΝΟΛΟ ΔΕΔΟΜΕΝΩΝ (DATASET)

Τα δεδομένα (dataset) που έχουμε είναι μια συλλογή excel εγγράφων που το καθένα αντιπροσωπεύει ένα MOOC. Σε κάθε έγγραφο έχουμε διάφορες συνεδρίες (sessions) δηλαδή instances που 2 μαθητές με τη βοήθεια ενός πράκτορα (agent) προσπαθούν να βρουν λύση στο ερώτημα που τους δόθηκε. Το ερώτημα είναι ίδιο σε όλες τις συνεδρίες (sessions) του συγκεκριμένου MOOC.

Κάθε γραμμή στο excel έγγραφο αντιπροσωπεύει είτε ένα μήνυμα είτε το τέλος του session όταν αυτή η γραμμή είναι κενή. Η τελική απάντηση σε κάθε session βρίσκετε στη πρώτη γραμμή της αρχής του session στο αρχείο.

2.1 Η ΔΟΜΗ ΕΝΟΣ SESSION

Στο κάθε session μπαίνουν πάντα 2 μαθητές και τους γίνεται μια ερώτηση. Σε κάθε session υπάρχει και ένας πράκτορας (agent)(που είναι chat bot συνήθως ονομάζεται τιμ στο συγκεκριμένο MOOC) που τους βοηθάει με διάφορα στοχευμένα ερωτήματα να φτάσουν σε μια λύση.

Στην αρχή του session, ο agent καλωσορίζει τους 2 μαθητές** και στέλνει ένα default μήνυμα*, συνήθως και το κάθε άτομο καλωσορίζει τον συμπαίκτη του και ξεκινάν να βρουν λύση. Έπειτα αφού υποβάλουν τη λύση ανάλογα με το MOOC ο agent θα τους πει "Συγχαρητήρια! Η ομάδα σας υπέβαλε επιτυχώς την απάντηση στο υπό συζήτηση θέμα."

(*) "Τώρα που έχετε συνδεθεί και οι δυο, μπορούμε να ξεκινήσουμε! Σε αυτή τη συνεργατική δραστηριότητα, καλείστε να παρουσιάσετε μια κοινή απάντηση στο υπό συζήτηση θέμα που απεικονίζεται στο επάνω αριστερά μέρος του παραθύρου. Η δραστηριότητα μπορεί να ξεκινήσει με τις απαραίτητες συστάσεις μεταξύ μας. Ας ξεκινήσω εγώ. Με λένε [Agent] και ο αγαπημένος μου τομέας ενδιαφέροντος είναι η συνεργατική μάθηση."

(**)"Καλώς ήρθες [first_name_player_1]. Θα πρέπει να περιμένουμε μέχρι να συνδεθεί και ο συνεργάτης σου στη δραστηριότητα.

Καλώς ήρθες [first_name_player_2]. Θα πρέπει να περιμένουμε μέχρι να συνδεθεί και ο συνεργάτης σου στη δραστηριότητα."

2.1.1 Η δομή ενός μηνύματος

Όλα αυτά τα θεωρούμε ως μηνύματα. Κάθε μήνυμα είναι μία γραμμή σε αρχείο Excel.

Αυτή η γραμμή περιέχει διάφορα κελία που μας παρέχουν χρήσιμες πληροφορίες που θα μας βοηθήσουν αργότερα στην Ανάλυση Κειμένου.

1. sec : Το δευτερόλεπτο που καταχωρήθηκε το μήνυμα (ώρα καταχώρησης - ώρα έναρξης session) (Integer)
2. timestamp : ώρα καταχώρησης του μηνύματος (DateTime)
3. player_nick : το όνομα ενός μαθητή ή του agent (δηλαδή [first_name_player_1] / [first_name_player_2] / Agent) που έγραψε το μήνυμα (String)
4. player_id : ισούται με player_1_id ή player_2_id ή '0' αν το μήνυμα το έγραψε ο agent (Integer)
5. message_type : Τύπος του μηνύματος. Τα απλά μηνύματα είναι '1' (Integer)
6. the_message : Το μήνυμα (String)
7. message_length : ο αριθμός των λέξεων του μηνύματος (Integer)
8. question_id : Είναι το ερώτημα του agent το οποίο απαντάτε με αυτό το μήνυμα (String)
9. question_direction : (first_name_player_1/first_name_player_2/both) για ποιον προορίζεται το μήνυμα του agent (String)
10. feedback_reason : είναι ≥0 αν πρόκειται για απάντηση σε ερώτηση του agent (Integer)
11. session_id : το id του συγκεκριμένου session(Integer, μόνο στη κενή γραμμή)
12. agent_id: το id του agent (είναι το ίδιο σε όλο το έγγραφο)(Integer, μόνο στη κενή γραμμή)
13. first_name_player_1 : Όνομα 1ου μαθητή (String, μόνο στη κενή γραμμή)
14. last_name_player_1 : Επώνυμο 1ου μαθητή (String, μόνο στη κενή γραμμή)
15. player_1_id : id 1ου μαθητή (Integer, μόνο στη κενή γραμμή)
16. first_name_player_2 : Όνομα 2ου μαθητή (String, μόνο στη κενή γραμμή)
17. last_name_player_2 : Επώνυμο 2ου μαθητή (String, μόνο στη κενή γραμμή)
18. player_2_id : id 2ου μαθητή (Integer, μόνο στη κενή γραμμή)

2.1.2 Τύποι μηνυμάτων

Κάθε μήνυμα έχει και ένα τύπο(message_type). Οι μαθητές χρησιμοποιούν διαφορετικούς τύπους μηνυμάτων σε σχέση με τον Agent

- 0 : [Μαθητές] Λύση

- 1: [Μαθητές] Χαιρετισμός ή απλή συνομιλία
- 2: [Agent] "Καλώς ήρθες [Username]..." από τον Agent
- 3: [Agent] Το ίδιο μήνυμα πάντα "τώρα που έχετε συνδεθεί και οι δυο, μπορούμε να ξεκινήσουμε!..."
- 4: [Agent] "Καλώς ήρθες πάλι πίσω [Username]" από τον Agent
- 5: [Agent] "Φαίνεται ότι ο χρήστης [Username] αποσυνδέθηκε από τη δραστηριότητα" από τον Agent
- 6: [Agent] "Συγχαρητήρια! Η ομάδα σας υπέβαλε επιτυχώς την απάντηση στο υπό συζήτηση θέμα." από τον Agent
- 7: [Agent] Το ίδιο μήνυμα "Το πρόβλημα μας κάνει να σκεφτόμαστε ζευγάρια από μία ελληνική λέξη.." λογικά είναι για βοήθεια
- 8: Δεν υπάρχει 8
- 9: [Agent] Ο Agent Λογικά βοηθάει τους χρήστες βάζοντας τους να εξηγήσουν περισσότερο την απάντηση τους ώστε να βρουν την λύση
- 10: [Μαθητές] Απαντήσεις μαθητών στα message type 9 του Agent
- 11: [Agent] Βοηθητικό μήνυμα από Agent που προορίζετε για κάθε χρήστη ξεχωριστά
- 12: [Agent] Ο Agent πιάνει λέξεις στις απαντήσεις των χρηστών και τους ρωτά πράγματα πάνω σε αυτό

2.2 Ο ΠΡΑΚΤΟΡΑΣ (AGENT)

Ο λόγος ύπαρξης του πράκτορα Agent είναι να βοηθήσει τους μαθητές να φτάσουν σε μια λύση. Το κάνει θέτοντάς ερωτήματα που έχουν στόχο να απαντηθούν αλλά και να ξεκινήσει διάλογος μεταξύ των μαθητών. Στα δεδομένα μας το chat bot έχει το όνομα [player_nick] = “Agent” και [player_id] = 0 αλλά οι μαθητές προσπαθούν μάταια να τον καλέσουν με το όνομα Τιμ για να τους βοηθήσει.

Στον Agent έχει δοθεί μια λίστα με trigger words και σε κάθε μήνυμα των μαθήματων προσπαθεί να βρει έστω μια από αυτές τις λέξεις. Όταν τη βρει, κάνει μια παρέμβαση(παρέμβαση ονομάζουμε όλα τα μηνύματα του Agent) και στέλνει μια από τις ερωτήσεις του. Τα ερωτήματα ανά MOOC είναι συγκεκριμένα και μπορούν να εμφανιστούν μόνο μια φορά ανά session.

2.3 ΠΟΛΙΤΙΚΗ ΕΚΚΑΘΑΡΙΣΗΣ ΣΥΝΟΛΟΥ ΔΕΔΟΜΕΝΩΝ

Το κάθε αρχείο Excel έρχεται με διάφορα sessions και μηνύματά που δεν βοηθάνε στη τελική λύση. Έχουμε πάρει την απόφαση να κρατήσουμε όλα μήνυμα ανά session αλλά διαγράφουμε όλο το session αν τηρεί έστω ένα από τα παρακάτω.

Διαγραφή session αν:

- ο ένας χρήστης έχει αποσυνδεθεί για το περισσότερο διάστημα του session
- μόνο ο ένας χρήστης προσπάθησε να βρει τη λύση
- δεν δόθηκε λύση ή δόθηκαν άσχετοι χαρακτήρες ώστε να τελειώσει το session

Ανά session είναι πιθανόν να βρούμε μήνυμα που δεν παίζουν ρολό στη λύση του MOOC. Αφήσαμε αυτά τα μηνύματά διότι δεν παίζουν ρολό στο topic modeling που θα κάνουμε αλλά θα τα χρειαστούμε στη στατιστική ανάλυση που θα κάνουμε παρακάτω. Επιπλέον χάνετε η ροή του διαλόγου με τη διαγραφή αυτών των μηνυμάτων. Επομένως απλά τα σημειώνουμε με κίτρινο αν πληροί έστω ένα από τα παρακάτω

Σήμανση μηνύματος αν:

- δεν βοηθάει προς τη λύση του προβλήματος
- χαιρετισμοί και αποχαιρετισμοί
- μηνύματα με λίγα γράμματα που δεν σχηματίζουν λέξη (πχ "χδ")

Τέλος επισημαίνουμε με κόκκινο χρώμα ολόκληρες γραμμές ή συγκεκριμένα κελιά οπού υπάρχει κάποιο σφάλμα. Για παράδειγμα:

- λάθος όνομα σε παρέμβαση του Agent

9:2ε	Agent	0	3	Τώρα που έχετε συνδεθεί και οι δυο, μπορούμε να ξ
0:03	Θεοδώρα	3025	1	Καλησπερα
0:06	Θεοδώρα	3025	1	Ξεκινάμε?
0:11	KONSTANTINOS	2994	1	γ
1:31	KONSTANTINOS	2994	1	πιστευω πως η λυση ειναι τα λεξικα
1:32	Agent	0	9	Αναφέρατε το λεξικό. Σκεφτείτε το εξής: Αν θέλετε ν
2:52	KONSTANTINOS	2994	10	οτι το λεξικο μου πρεπει να εχει λιγα ζευγη
2:52	Agent	0	11	Θεοδώρα μήπως μια δομή που συνδυάζει λίστα και
2:52	Θεοδώρα	3025	1	Ναι λεξικο ειναι η απαντηση
3:06	Agent	0	5	Φαίνεται ότι ο χρήστης Σταυρούλα αποσυνδέθηκε α
3:26	Θεοδώρα	3025	1	Ποια ειναι η Σταυρουλα?

Εικόνα 4: Λάθος όνομα μαθητή

- Κενά σημαντικά κελιά όπως (player_nick, player_id, message_type, the_message). Παρατηρήθηκε ότι αυτό συνήθως σημαίνει όταν ένας μαθητής αποσυνδεθεί και το σύστημα μάλλον δεν είχε καταγράψει γρήγορα ότι συνδέθηκε ξανά με αποτέλεσμα το μήνυμα του μαθητή να μην έχει “player_nick”, “player_id”.

Ούτε τα κόκκινα κελιά δεν τα διαγράφουμε διότι έχουν ακόμα κάποια πληροφορία μέσα τους και τα θεωρούμε μηνύματα των μαθητών. Επισημαίνουμε αυτές τις γραμμές ώστε να γίνει ευκολότερη η ανάλυση κάποια άλλη φορά που θα χρειαστούμε αυτά τα δεδομένα.

ΚΕΦΑΛΑΙΟ 3: ΑΝΑΛΥΣΗ ΚΕΙΜΕΝΟΥ

ΑΝΑΛΥΣΗ ΚΕΙΜΕΝΟΥ

Σε αυτό το σημείο θα εξετάσουμε τα 2 έγγραφα Excel ξεχωριστά ώστε να βρούμε στατιστικές διάφορες καθώς στο ένα έχουμε μαθητές Πληροφορικής (στη συνέχεια ο πληθυσμός αυτός θα αναφέρεται ως "SPOC" = Small Private Online Course) και στο άλλο εκπαιδευόμενους με ποικίλο υπόβαθρο γνώσεων (στη συνέχεια ο πληθυσμός αυτός θα αναφέρεται ως "MOOC").

Πριν αρχίσει η ανάλυση θα χρειαστεί να θέσουμε κάποιους κανόνες ώστε να βγουν αποτελέσματα που θα μας βοηθήσουν.

- Τα κενά μηνύματα δεν συμβάλλουν στα τελικά στατιστικά δεδομένα (τα περισσότερα έχουν σημειωθεί με κόκκινο μέσα στο Excel).
- Στο έγγραφο θεωρούμε ότι το session τελειώνει όταν βρούμε κενή γραμμή διότι η τελική απάντηση βρίσκεται στη πρώτη γραμμή του κάθε session.
- Τη χρονική διάρκεια ενός session μπορούμε να την πάρουμε από την απάντηση της ομάδας (δηλαδή το τελευταίο μήνυμα του session) από το κελί “sec”.
- Στις παρεμβάσεις το Agent δεν μετρώ τις αρχικές που είναι χαιρετισμός προς τους μαθητές, τις συνδέσεις και αποσυνδέσεις. Άρα μετρώ μόνο τα message_type 6, 7, 9, 11 και 12.
- Οι targeted παρεμβάσεις από Agent έχουν message_type 9 ή 11.
- Οι παρεμβάσεις προς όλη την ομάδα από Agent έχουν message_type 6 ή 7 ή 12.
- Η πρώτη παρέμβαση του Agent είναι μετά τους χαιρετισμούς και δεν υπολογίζει τις συνδέσεις/αποσυνδέσεις ως παρέμβαση.
- Η τελευταία παρέμβαση του Agent είναι πριν την λύση του προβλήματος και δεν υπολογίζει τις συνδέσεις/αποσυνδέσεις ως παρέμβαση.
- Στο ερώτημα “ Χρόνος που χρειάστηκε για να δοθεί απάντηση “ μετράμε την πρώτη απάντηση που θα δοθεί και όχι τις υπόλοιπες πάνω στην παρέμβαση.
- Στην ερώτηση “Μετά από ποσά μηνύματα εμφανίστηκε η τελευταία παρέμβαση του Agent” δεν μετράμε τις παρεμβάσεις με message_type=6 διότι το sec είναι ίσο με 0.
- Στα ερωτήματα για τις παρεμβάσεις που απαντήθηκαν, ο μόνος τρόπος να τις βρούμε είμαι με το να συγκρίνουμε τα strings διότι δεν έχουμε κάποιο message id που θα μας βοηθούσε.

Επίσης, θέλουμε να δούμε από τη λίστα ερωτημάτων που έχει ο Agent ποιες παρεμβάσεις εμφανίζονται συχνά. Γνωρίζουμε ότι το κάθε ερώτημα μπορεί να εμφανιστεί το πολύ μια φορά σε κάθε session με διαφορετικό [question_id]. Αρά αυτό που κάνουμε είναι να παίρνουμε όλα [question_id] από τις παρεμβάσεις με message_type ίσο με 7,9,11 και 12.

3.1 ΑΠΟ ΠΛΗΘΥΣΜΟ SPOC

Στόχος	Μέση Τιμή	Τυπική Απόκλιση
Αριθμός μηνυμάτων μαθητών ανά session	40,41	28,04
Αριθμός παρεμβάσεων του Agent	6	2,25

Στόχος	Μέση Τιμή	Τυπική Απόκλιση
Αριθμός στοχοποιημένων παρεμβάσεων που απαντήθηκαν	0,77	0,56
Αριθμός ομαδικών παρεμβάσεων που απαντήθηκαν	0,33	0,47
Μέγεθος σε λέξεις που είχαν οι απλές απαντήσεις των μαθητών	7,48	8,25
Χρόνος που χρειάστηκε για να δοθεί απάντηση (δευτερόλεπτα)	100,63	107,41
Μετά από ποσά μηνύματα εμφανίστηκε η πρώτη παρέμβαση του Agent	8,57	4,15
Μετά από ποσά μηνύματα εμφανίστηκε η τελευταία παρέμβαση του Agent	31,41	20,33
Μέγεθος σε λέξεις που είχαν οι τελικές απαντήσεις των μαθητών	88,08	44,17
Χρονική διάρκεια session	1879,84	1122,6
Ποσοστό μηνυμάτων από τον Agent	15%	7%
Μέσος Όρος Μεγέθους Μηνύματος Agent ανά session	32,13	7,02
Ποσοστό μηνυμάτων από τους μαθητές	85%	7%
Μέσος Όρος Μεγέθους Μηνύματος Μαθητή ανά session	14,31	6,40

Πίνακας 1: Αποτελέσματα στόχων του πληθυσμού SPOC

Question ID	Πιθανότητα εμφάνισης σε session
0	88%
1	7%
2	95%
3	72%
4	77%
5	17%
6	58%
7	36%
8	84%
9	33%

Πίνακας 2: Πιθανότητα εμφάνισης παρέμβασης σε συνεδρία του πληθυσμού SPOC

3.2 ΑΠΟ ΠΛΗΘΥΣΜΟΥ MOOC

Στόχος	Μέση Τιμή	Τυπική Απόκλιση
Αριθμός μηνυμάτων μαθητών ανά session	53,47	54,89
Αριθμός παρεμβάσεων του Agent	5,6	2,24
Αριθμός στοχοποιημένων παρεμβάσεων που απαντήθηκαν	1,66	0,71
Αριθμός ομαδικών παρεμβάσεων που απαντήθηκαν	0,28	0,44
Μέγεθος σε λέξεις που είχαν οι απλές απαντήσεις των μαθητών	11,28	8,8
Χρόνος που χρειάστηκε για να δοθεί απάντηση (δευτερόλεπτα)	143,54	153,3
Μετά από ποσά μηνύματα εμφανίστηκε η πρώτη παρέμβαση του Agent	8,76	3,65
Μετά από ποσά μηνύματα εμφανίστηκε η τελευταία παρέμβαση του Agent	35,56	26,08
Μέγεθος σε λέξεις που είχαν οι τελικές απαντήσεις των μαθητών	108,98	58,89
Χρονική διάρκεια session	2107,01	1366,84
Ποσοστό μηνυμάτων από τον Agent	13%	6%
Μέσος Όρος Μεγέθους Μηνύματος Agent ανά session	34,78	7,52
Ποσοστό μηνυμάτων από τους μαθητές	87%	6%
Μέσος Όρος Μεγέθους Μηνύματος Μαθητή ανά session	13,2	5,34

Πίνακας 3: Αποτελέσματα στόχων του πληθυσμού MOOC

Question ID	Πιθανότητα εμφάνισης σε session
0	85%
1	10%
2	95%
3	66%
4	81%
5	18%
6	56%

Question ID	Πιθανότητα εμφάνισης σε session
7	35%
8	83%
9	32%

Πίνακας 4: Πιθανότητα εμφάνισης παρέμβασης σε συνεδρία του πληθυσμού MOOC

3.3 ΑΞΙΟΣΗΜΕΙΩΤΑ ΣΥΜΠΕΡΑΣΜΑΤΑ ΑΝΑΛΥΣΗΣ

Από τα παραπάνω ευρήματα μπορούμε να βρούμε κάποιες διάφορες μεταξύ των πληθυσμού SPOC και του πληθυσμού MOOC. Αρχικά, ο αριθμός μηνυμάτων των MOOC είναι αρκετά μεγαλύτερος και με τη διπλάσια τυπική απόκλιση. Αυτό μπορεί να οφείλεται στο ότι άτομα στο MOOC έρχονται από διάφορους κλάδους που δεν έχουν το ίδιο επίπεδο γνώσεων πάνω στη Πληροφορική. Οπότε είτε και οι δύο προσπαθούν πολύ για τη λύση είτε ο ένας εξηγεί στον άλλο τι προσπαθούν να κάνουν.

Το ίδιο μπορεί να θεωρηθεί για το μέγεθος και το χρόνο που χρειάστηκαν οι μαθητές του πληθυσμού MOOC που είναι υψηλότερα σε σχέση με των μαθητών του πληθυσμού SPOC που πιθανόν να είχαν ένα ξεκάθαρο στόχο στο μυαλό τους και δεν είχαν χρόνο να απαντήσουν στα ερωτήματα. Επιπλέον οι μαθητές του SPOC έχουν μικρότερη πιθανότητα να απαντήσουν σε παρέμβαση (SPOC: 0,77 < MOOC: 1,66) πράγμα που ισχυροποιεί περισσότερο την υπόθεση μου.

Τέλος η πιθανότητα εμφάνισης κάποιας συγκείμενης ερώτησης από τον Agent είναι όμοια εκτός από το question_id = 3 που έχει περίπου 10% μεγαλύτερη πιθανότητα εμφάνισης στους μαθητές του SPOC.

ΚΕΦΑΛΑΙΟ 4: ΘΕΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ (TOPIC MODELING)

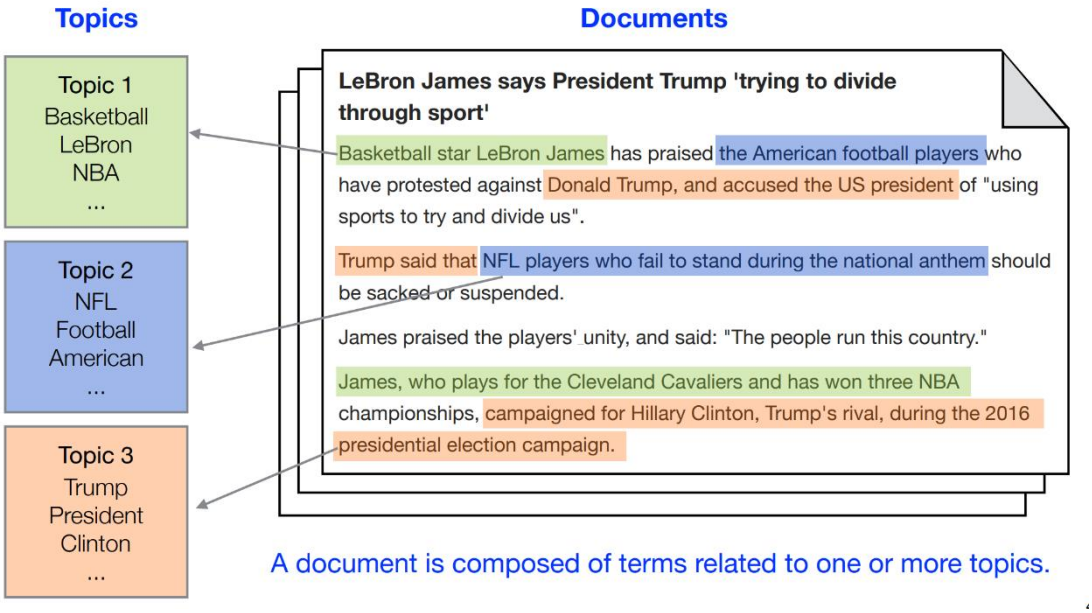
ΘΕΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ (TOPIC MODELING)

4.1 ΕΙΣΑΓΩΓΗ ΣΤΟ TOPIC MODELING

Κάθε απάντηση στο MOOC, πέρα από λανθασμένη/μερικώς σωστή/σωστή, μας προσφέρει περισσότερες πληροφορίες που μπορούμε να αντλήσουμε ώστε να βγάλουμε περεταίρω συμπεράσματα. Το Topic Modeling (Μοντελοποίηση Θεμάτων) μας βοηθάει σημαντικά σε αυτό.

Η γλώσσα είναι περιπλοκή και δεν ακολουθούν όλοι τους ιδίους κανόνες που χρησιμοποιούμε εμείς. Υπάρχουν διακυμάνσεις στην οργάνωση και μορφή της πρότασης και γενικότερα του λογού, η Μοντελοποίηση Θεμάτων έχει στόχο να αναγνωρίσει πρότυπα και να ανακαλύψει μοτίβα. Αφήνει "πίσω" τις θεωρείς μας για το πως πρέπει να ορίζονται τα πράγματα και παρατηρεί γενικότερα θέματα.

Η Μοντελοποίηση Θεμάτων ανήκει στον κλάδο προβλημάτων μάθησης χωρίς επίβλεψη (Unsupervised Learning), δηλαδή στη περίπτωση μας δεν έχουμε δώσει ετικέτες στα δεδομένα μας επειδή στόχο έχουμε να βρούμε θεματικές χωρίς να είναι επηρεασμένες από εμάς. Το μόνο που χρειάζεται να γνωρίζει ο αλγόριθμος είναι το πόσες θεματικές χρειάζεται να βρει, έπειτα θα προσπαθήσει μονός του να βρει συσχετισμούς στο σύνολο δεδομένων χωρίς να γνωρίζει αν υπάρχουν και πόσοι είναι.



Εικόνα 5: Λέξεις ανά θέμα, Topic Modeling with Scikit-learn (Greene, 2017)

Εν συντομία, κάθε μοντέλο θεμάτων είναι βασισμένο στην εξής υπόθεση (Blei, 2003):

- κάθε έγγραφο αποτελείται από ένα μείγμα θεμάτων
- κάθε θέμα αποτελείται από μια συλλογή λέξεων.

4.1.1 Επιλογή Topic Modeling ή Topic Classification

Στο κλάδο της ανάλυσης θεμάτων έχουμε 2 βασικές τεχνικές:

- Topic Modeling (Μοντελοποίηση Θεμάτων)
- Topic Classification (Κατηγοριοποίηση Θεμάτων)

Μπορεί να φαίνεται ότι και οι δυο φράσεις έχουν το ίδιο νόημα αλλά υπάρχει μια σημαντική διαφορά το topic modeling είναι προβλήματα μάθησης χωρίς επίβλεψη (Unsupervised Learning) ενώ το topic classification (κατηγοριοποίηση θεμάτων) είναι προβλήματα μάθησης με επίβλεψη (Supervised Learning).

Αυτό σημαίνει ότι στη μοντελοποίηση δεν χρειάζεται να αναθέσουμε εμείς σε κάθε ένα από τα έγγραφα ένα θέμα. Είναι ένας απλός και γρήγορος τρόπος να αναλύσεις τα δεδομένα σου αλλά τα τελικά θέματα που θα σου βγάλει μπορεί να μην βγάζουν κάποιο νόημα. Το βασικό μειονέκτημα είναι ότι χρειάζεται πολλά έγγραφα για να βγάλει σημαντικά αποτελέσματα

Αντιθέτως, η κατηγοριοποίηση χρειάζεται να εκπαιδευτεί με έγγραφα που έχουμε κατηγοριοποιήσει εμείς *πχ σε έγγραφα από μαθήματα Γυμνάσιου {Έγγραφο 1: Ιστορία, Έγγραφο 2: Γυμναστική, Έγγραφο 3 Φυσική, Έγγραφο 4: Ιστορία}*. Αυτό όπως μπορούμε να καταλάβουμε παίρνει χρόνο και ανθρώπινη προσπάθεια, αρά οι αλγόριθμοι είναι τόσο αμερόληπτοι στη κατηγοριοποίηση νέων εγγράφων όσο και οι δημιουργοί των ετικετών (θεμάτων) στο αρχικό σύνολο δεδομένων (dataset). Από την άλλη μπορούμε να δούμε πως νέα έγγραφα κατηγοριοποιούνται σε μια λίστα από προκαθορισμένα θέματα.

Τέλος εμείς αποφασίσαμε να χρησιμοποιήσουμε topic modeling πρώτον γιατί η υλοποίηση είναι γρήγορη και μπορούμε ευκολά να βρούμε περισσότερα έγγραφα από MOOCs. Δευτέρων επειδή προσπαθούμε βρούμε τις μικρές διακυμάνσεις σε μια πλειάδα από παρόμοιες απαντήσεις που δεν είναι αντιληπτές από τον άνθρωπο. Αν ήμασταν μια επιχείρηση που μια από τις διαφημιζόμενες δυνατότητες μας ήταν η αυτόματη αξιολόγηση τότε φυσικά θα επενδύαμε χρόνο και κεφάλαιο σε μια σωστή κατηγοριοποίηση θεμάτων.

4.2 ΤΕΧΝΙΚΕΣ TOPIC MODELING

Μπορούμε να αντιμετωπίσουμε το πρόβλημα του topic modeling με διάφορες τεχνικές που έχουν αναπτυχθεί ανά τα χρονιά. Οι δυο κυρίες κατηγορίες προσεγγίσεων είναι:

1. **Πιθανοτική προσέγγιση:** Θεωρούμε πως το κάθε έγγραφο αποτελείτε από λίγα θέματα και δίνουμε σε λέξεις και έγγραφα τη πιθανότητα συσχέτισης για κάθε θέμα.
2. **Γραμμική Άλγεβρα:** Με μεθόδους γραμμικές άλγεβρας μπορεί ένας πίνακας με {έγγραφο - λέξη} να γίνει αποσύνθεση σε μικρότερους πίνακες και αυτό να το ερμηνεύσουμε ως μοντέλο θέματος. (Xu, 2018)

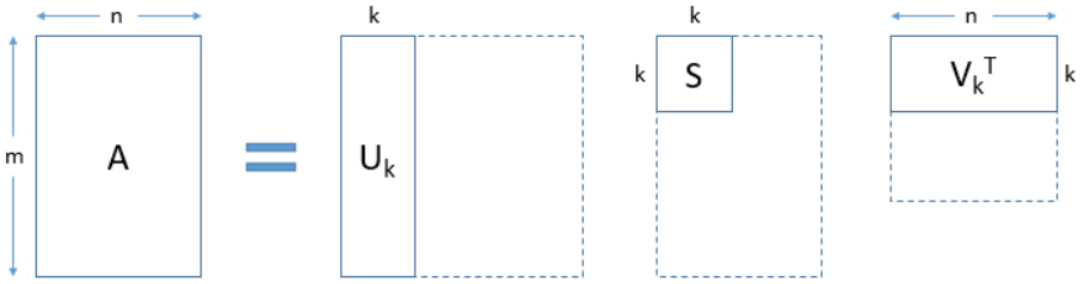
4.2.1 Latent Sematic Analysis (LSA)

Αποκαλείτε και Latent Semantic Indexing (LSI) όταν το χρησιμοποιούμε στο κλάδο Ανάκτησης Πληροφοριών (Information Retrival). Δημιουργήθηκε από τους Landauer και Dumais το 1997 ως μια στατιστική τεχνική που συσχετίζει σημασιολογικά συνδεδεμένους όρους με μια συλλογή εγγράφων (corpora/corpus). Το LSA χρησιμοποιεί αποσύνθεση ή αλλιώς παραγοντοποίηση ιδιαζουσών τιμών (singular value decomposition, ή SVD). Το σκεπτικό είναι ότι παρόμοιοι όροι τείνουν να χρησιμοποιούνται στο ίδιο πλαίσιο και ως εκ τούτου τείνουν να συνυπάρχουν περισσότερο. Για παράδειγμα ο όρος "δεδομένα" θα συνδεθεί με όρους του ίδιου πλαισίου δηλαδή με όρους όπως "υπολογιστής", "επεξεργασία" και "μαθηματικά" που δεν χρειάζεται να σημαίνουν το ίδιο όπως το "δεδομένα".

Το LSA όπως οι περισσότεροι αλγόριθμοι topic modeling παίρνει ως όρισμα ένα πίνακα έγγραφο – όρος (document - term matrix) δηλαδή κάθε γραμμή αντιπροσωπεύει ένα έγγραφο και κάθε στήλη αντιπροσωπεύει ένα όρο. Αυτός ο πίνακας μας δείχνει αν μία λέξη X υπάρχει σε ένα έγγραφο Y και θα αναλύσουμε σε επόμενο κεφάλαιο το πως δημιουργούμε ένα τέτοιο πίνακα. Το μόνο που θα χρειαστεί να προσέξουμε είναι, να μην χρησιμοποιήσουμε απλό μέτρημα των λέξεων για το document - term matrix αλλά να επιλέξουμε tf-idf. Το Tf-idf (Term Frequency – Inverse Document Frequency) που υπολογίζει ένα βάρος βάση της συχνότητας εμφάνισης στο συγκεκριμένο έγγραφο και στο σύνολο των εγγράφων (corpus).

Όταν το document - term matrix είναι έτοιμο τότε μπορούμε να εκτελέσουμε το LSA αφού το έχουμε δώσει τον αριθμό k των θεμάτων που θέλουμε να βρει. Ο αλγόριθμος θα προσπαθήσει να μειώσει το μέγεθος του document - term matrix σε ένα k x k πίνακα χρησιμοποιώντας παραγοντοποίηση ιδιαζουσών τιμών (singular value decomposition, ή SVD)

$$A = USV^T$$



Εικόνα 6: LSA Αποσύνθεση SVD (Prateek, 2018)

Το LSA είναι μία από τις παλαιότερες και πιο διαδεδομένες τεχνικές Μοντελοποίησης Θεμάτων. Είναι γρήγορο και εύκολο στην υλοποίηση του αλλά έχει ως κύριο μειονέκτημα ότι χρειάζεται μεγάλο όγκο δεδομένων για να πάρουμε καλά αποτελέσματα. Αργότερα παρουσιάστηκε μια πιθανολογική παραλλαγή του LSA, το PLSA που έφερε βελτιώσεις και έπειτα ο Blei παρουσίασε το LDA που είναι Bayesian εκδοχή του PLSA.

4.2.2 Latent Dirichlet Allocation (LDA)

Παρουσιάστηκε το 2003 από τους David Blei, Andrew Ng και Michael O. Jordan. Είναι αλγόριθμος μάθησης χωρίς επίβλεψη και κύριο παράδειγμα λειτουργίας είναι η μοντελοποίηση θεμάτων. Ο LDA είναι η Bayesian έκδοση του pLSA (Probabilistic Latent Semantic Analysis, χρησιμοποιεί μια πιθανολογική μέθοδο αντί για το SVD για την αντιμετώπιση του προβλήματος) .

Συγκεκριμένα, χρησιμοποιεί προηγμένες τεχνικές Dirichlet για το θέμα του εγγράφου και τις κατανομές λέξεων-θέματος, προσδίδοντάς του την καλύτερη γενίκευση.

Έχουμε δυο κατανομές, μια έγγραφο - θέμα (document-topic) και μια λέξη – θέμα (word-topic) και χρησιμοποιώντας Dirichlet κατανομές (επεξήγηση Dirichlet) μπορούμε να βρούμε τη πιθανότητα να υπάρχει ένα θέμα σε ένα έγγραφο πχ Document 1: 8% topic A, 85% topic B, 7% topic C .

Όπως αναφέραμε ο αλγόριθμος LDA ανήκει στη κατηγορία προβλημάτων μάθησης χωρίς επίβλεψη. Αυτό σημαίνει ότι είναι πολύ πιθανό να μας βγάζει θεματικές που δεν ταιριάζουν ακριβώς με την αντίληψη του ανθρώπου, όμως αυτό μπορεί να είναι καλό διότι βρίσκει κάποιες συσχετίσεις λέξεων που εμείς δεν θα τις βρίσκαμε. Στην ουσία ο αλγόριθμος LDA είναι μια τεχνική μείωσης τα δεδομένων, βρίσκει τις πολύ κοινές συσχετίσεις στη συλλογή εγγράφων (corpus) και τις αντιστοιχεί σε κοινά θέματα. Αυτό όμως έχει ως αποτέλεσμα να μην βρει θεματικές που εμφανίζονται ελάχιστα και ασυνήθιστες φράσεις από θεματικές δεν θα καταγράφουν. Μπορούμε όμως να εξετάσουμε τον αλγόριθμο όχι μόνο με μεμονωμένες λέξεις αλλά και με φράσεις δύο (bigrams) και τριών (trigrams) λέξεων. Έτσι θα ελέγξουμε πως συμπεριφέρεται ο LDA με φράσεις των εγγράφων.

Βήματα LDA(Sarkar, 2019):

1. Αρχικοποιεί τις απαραίτητες παραμέτρους.
2. Για κάθε έγγραφο, δίνει τυχαία κάθε λέξη σε ένα από τα K θέματα.
3. Κάνει τη παρακάτω διαδικασία επαναληπτικά i φορές. Για κάθε έγγραφο D, Για κάθε λέξη W σε ένα έγγραφο, Για κάθε θέμα T:
    1. Υπολογίζει P(T|D), δηλαδή το ποσοστό των λέξεων στο D που έχουν ανατεθεί στο θέμα T.
    2. Υπολογίζει P(W|T), δηλαδή το ποσοστό των αναθέσεων του θέματος T πάνω σε όλα τα έγγραφα που έχουν τη λέξη W.

3. Αναθέτει ξανά τη λέξη W με το θέμα T με τη πιθανότητα P(T|D) x P(W|T) λαμβάνοντας υπόψη όλες τις άλλες λέξεις και τις αναθέσεις των θεμάτων τους.

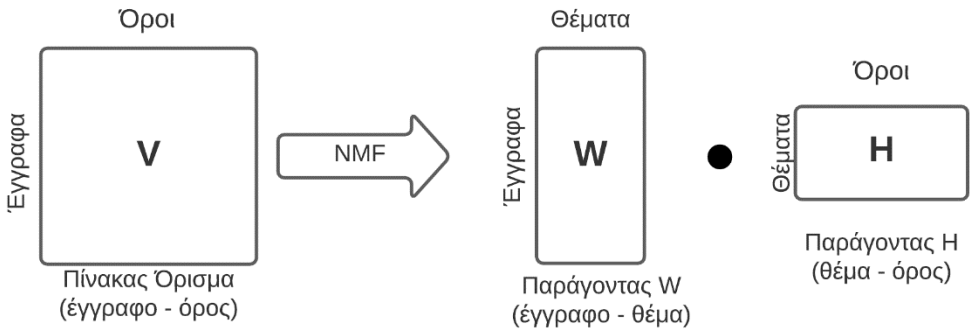
Παράμετροι του LDA:

- alpha: με μικρότερο alpha ο αλγόριθμος θα αναθέσει λιγότερα θέματα ανά έγγραφο ενώ με μεγαλύτερο alpha θα αναθέσει περισσότερα θέματα ανά έγγραφο
- beta: με μικρότερο beta θα χρησιμοποιήσει λιγότερες λέξεις για να μοντελοποίηση ένα θέμα ενώ με μεγαλύτερο beta θα χρησιμοποιήσει περισσότερες λέξεις αρά τα θέματα θα μοιάζουν περισσότερο μεταξύ τους
- K: ο αριθμός των θεμάτων
- random_state: Ο LDA είναι πιθανολογικό μοντέλο και έτσι κάθε φορά που εκτελείτε μας βγάζει άλλο αποτέλεσμα. Αν δεν θέλουμε κάτι τέτοιο τότε βάζουμε ένα συγκεκριμένο αριθμό και δεν το αλλάζουμε.

4.2.3 Non-negative Matrix Factorization (NFM)

Το Non-negative Matrix Factorization(NMF ή NNMF) είναι άλλη μία τεχνική μάθησης χωρίς επίβλεψη άρα δεν δίνουμε ετικέτες στα έγγραφα που θα εκπαιδευτεί πάνω, το μόνο που χρειάζεται είναι ο αριθμός θεμάτων που θέλουμε. Στη πραγματικότητα το NMF είναι μια συλλογή από αλγορίθμους που χρησιμοποιούν γραμμική άλγεβρα ώστε να βρούνε τη κρυφή δομή σε δεδομένα. Γενικότερα προσπαθεί να μειώσει της διαστάσεις του εισερχόμενου πίνακα. Αυτό το πετυχαίνει με το να βρει 2 μη αρνητικούς πίνακες(non-negative matrix) W και H που αν πολλαπλασιαστούν μεταξύ τους θα βγάλουν έναν παρόμοιο πίνακα όπως το όρισμα μας V.

Δηλαδή : $V \approx W * H$



Εικόνα 7: NMF παραγοντοποίηση ενός document – term matrix

Για παράδειγμα αν έχουμε ως όρισμα ένα πίνακα με 500 γραμμές και 200 στήλες (500x200) που κάθε γραμμή είναι ένα έγγραφο και κάθε στήλη αντιπροσωπεύει έναν όρο του λεξιλογίου. Το μέγεθος του πίνακα W καθορίζεται από τον αριθμό των θεμάτων. Επομένως αν είχαμε k = 5 θέματα τότε ο πίνακας W θα έπρεπε να πάρει το σχήμα 200x5 (οροί - θέματα). Συνεχίζοντας, το μέγεθος του πίνακα H θα πρέπει να είναι 5x500 (θέματα - έγγραφα). Άρα αν πολλαπλασιάσουμε το W με το H βγάζουμε πίνακα με μέγεθος ίδιο με του ορίσματος (500x200)

Ο NMF θα ξεκινήσει με τυχαίες θετικές τιμές για να γεμίζει του πίνακες W και H. Έπειτα ο αλγόριθμος θα ενημερώνει τις τιμές μέχρι να μειωθεί το σφάλμα προσέγγισης ή μέχρι να τελειώσει ο μέγιστος αριθμός από επαναλήψεις.

Όταν τελειώσει η διαδικασία θα έχουμε τους δύο πίνακες W και H. Στον πίνακα W κάθε γραμμή αντιπροσωπεύει ένα έγγραφο και κάθε στήλη αντιπροσωπεύει το θέμα. Το κάθε κελί περιέχει ένα σταθμισμένο βάρος του έγγραφου ως προς το κάθε θέμα. Μπορούμε να βρούμε το πιο σχετικό θέμα για κάθε έγγραφο αν ταξινομήσουμε τις τιμές της στήλης σε φθίνουσα σειρά. Στον πίνακα H κάθε γραμμή αντιπροσωπεύει ένα θέμα και κάθε στήλη αντιπροσωπεύει τον όρο. Το κάθε κελί περιέχει ένα σταθμισμένο βάρος του όρου ως προς το κάθε θέμα. Μπορούμε να βρούμε του πιο σχετικούς όρους για κάθε θέμα αν ταξινομήσουμε τις τιμές της σειράς σε φθίνουσα σειρά.

Το NMF μπορεί να χρησιμοποιηθεί αντί του LDA σε περιπτώσεις που θέλουμε να παράγουμε μια πιο αραιή αναπαράσταση (sparse representation). Δηλαδή τα περισσότερα κελία του πίνακα θα έχουν τιμές κοντά στο μηδέν και έτσι βγάζει καλά αποτελέσματα σε μικρότερα έγγραφα με λιγότερες θεματικές. Επίσης από την έρευνα μας παρατηρήθηκε ότι είναι πιο γρήγορος από τον LDA.

Από την άλλη ο NMF δεν βγάζει σταθερά αποτελέσματα και με μία μικρή αλλαγή στις μεταβλητές του βγάζει διαφορετικά αποτελέσματα ειδικότερα για μεγαλύτερο αριθμό θεμάτων.

4.3 SHORT-TEXT TOPIC MODELING (STTM)

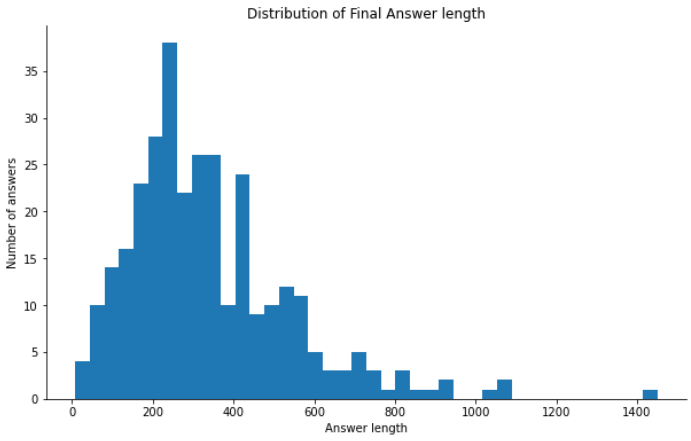
4.3.1 Περιορισμοί των γνωστότερων αλγορίθμων topic modeling για σύντομο κείμενο

Οι περισσότερες μελέτες έχουν γίνει σε έγγραφα με πολλές παραγράφους, άρθρα και εκθέσεις μαθητών. Δηλαδή εφαρμόζουν τους αλγορίθμους σε έγγραφα που υπάρχει μεγάλη ποικιλία από λέξεις. Είναι πολύ πιθανό ο συγγραφέας, στη προσπάθεια του να μην φαίνεται το κείμενο επαναλαμβανόμενο έχει χρησιμοποιήσει πολλές διαφορετικές λέξεις που έχουν το ίδιο νόημα. Για παράδειγμα οι όροι "άθληση", "άσκηση", "δραστηριότητα", "γυμναστική" έχουν το ίδιο νόημα οπότε μπορούν να εναλλάσσονται μεταξύ τους. Αυτή την ευελιξία δεν την έχουμε αλλά και δεν τη χρειαζόμαστε σε έγγραφα μικρού και μεσαίου μήκους. Έγγραφα όπως ερωτήσεις σε φόρουμ, τίτλοι, μηνύματα, tweets στο Twitter και στη περίπτωση μας μικρές απαντήσεις σε MOOC έχουν συγκεκριμένες διαφορές όπως:

- Η εναλλαγή των όρων με ίδιο μήνυμα σε μικρά κείμενα δεν γίνεται από ένα συγγραφέα αλλά από πολλούς. Ο καθένας χρησιμοποιεί και εκφράζεται με τον όρο που του ταιριάζει αλλά ο αλγόριθμος δεν μπορεί να καταλάβει εύκολα ότι έχουν το ίδιο νόημα διότι δεν έχει αρκετούς όρους από γύρω του.
- Με τον περιορισμένο αριθμό λέξεων και τη φύση αυτών των έγγραφων (όπως τίτλοι) είναι πολύ πιθανό να υπάρχει μόνο ένα θέμα στο έγγραφο. Αυτό μπορεί σε εμάς να φαίνεται πιο εύκολο αλλά για τους αλγόριθμους που είναι βασισμένοι στην υπόθεση ότι κάθε έγγραφο είναι μία μίξη θεμάτων τους επηρεάζει αρνητικά είτε σε επιδόσεις είτε σε αποτελέσματα.

```
import seaborn as sns
plt.figure(figsize=(10,6))
doc_lens = [len(d) for d in normalized]
plt.hist(doc_lens, bins = 40)
plt.title('Distribution of Final Answer length')
plt.ylabel('Number of answers')
plt.xlabel('Answer length')
sns.despine()
count = 0
for n in normalized:
    count = len(n)+count
aver = count/len(normalized)
print("Average message lenght:", str(aver))
```

```
Average message lenght: 339.70382165605093
```



Εικόνα 8: Μέσο μέγεθος εγγράφου από τα δεδομένα

Όπως αναφέραμε τα μικρά κείμενα περιέχουν ελάχιστες χρήσιμες λέξεις και έτσι είναι δύσκολο να εκλάβουμε πληροφορία για τη συνύπαρξη των λέξεων. Για αυτό το λόγο έχει παρατηρηθεί ότι οι γνωστοί αλγόριθμοι όπως LSA, LDA και NMF παράγουν μέτρια αποτέλεσμα διότι βασίζονται στη πληροφορία αυτή. Άρα προσπαθούμε να βρούμε τρόπο να παράγουμε αυτή τη πληροφορία ή να βρούμε μία άλλη τεχνική topic modeling που δεν χρησιμοποιεί πληροφορία συνύπαρξης λέξεων (word co-occurrence). Μια δημοφιλή τακτική είναι να μαζέψουμε μικρά κείμενα και να δημιουργήσουμε ένα μεγαλύτερο έγγραφο από τη μείξη τους, πχ όλα τα tweets που έχει κάνει ο χρήστης (Zhao, 2011). Αυτό όμως έχει ως αποτέλεσμα, τα θέματα που ανακαλύπτονται να μην είναι σωστά και να μην αντικατοπτρίζουν τη πραγματικότητα διότι πολλά άσχετα μικρά κείμενα μπορεί να έχουν προστεθεί στο μεγαλύτερο έγγραφο.

Μία άλλη γνωστή τακτική είναι να εκμεταλλευτούμε τον τεράστιο όγκο δεδομένων που υπάρχει στο διαδίκτυο και να εκπαιδεύσουμε ένα μοντέλο να βρίσκει τις

σημασιολογικές πληροφορίες (semantic information) και σχέσεις μεταξύ των λέξεων (Mikolov, 2013). Για παράδειγμα θα μπορούσαμε να εκπαιδεύσουμε ένα μοντέλο πάνω σε άρθρα που αφορούν τον ίδιο τομέα με τα δεδομένα μας. Αυτό δεν μας βοηθάει πάντα διότι υπάρχουν διαφορές μεταξύ άρθρου και μικρού κειμένου με αποτέλεσμα να εισάγουμε έξτρα θόρυβο.

4.3.2 Gibbs sampling for Dirichlet Mixture Model (GSDMM)

Στο χώρο του topic modeling ο αλγόριθμος Latent Dirichlet Allocation (LDA) είναι ο πιο γνωστός διότι είναι γρήγορος και αξιόπιστος. Όμως με την άνοδο του Twitter και του Reddit δημιουργούνται καθημερινά εκατομμύρια μικρά έγγραφα. Σε αυτά τα μικρά έγγραφα οι γνωστότεροι αλγόριθμοι βγάζουν μέτρια αποτέλεσμα επειδή δεν έχουν πολλά δεδομένα για να βρουν συσχετίσεις λέξεων. Ο αλγόριθμος collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model (GSDMM) παρουσιάστηκε από τους Yin και Wang το 2014 ως αντικαταστάτης του LDA για μικρά κείμενα διότι προσφέρει καλύτερα αποτελέσματα (Yin & Wang, 2014). Η βασική διαφορά του LDA και του GSDMM είναι ότι ο GSDMM θεωρεί πως κάθε έγγραφο έχει μόνο ένα θέμα και δεν είναι μια μείξη θεμάτων.

Ο αλγόριθμος, όπως και οι περισσότεροι του τομέα παίρνουν ως όρισμα τον αριθμό των θεμάτων και ένα πίνακα έγγραφο - ορός που μπορεί να δημιουργηθεί με διάφορους τρόπους. Συχνά δημιουργούμε τέτοιος πίνακας χρησιμοποιώντας τη τεχνική Term Frequency-Inverse Document Frequency(TF-IDF) που θα αναλύσουμε σε παρακάτω κεφάλαιο. Το πρόβλημα είναι πως σε μικρού μεγέθους κείμενα το TF-IDF δεν μας βγάζει καλά αποτέλεσμα διότι οι περισσότερες λέξεις εμφανίζονται μία φορά σε κάθε κείμενο αρά το TF=1. Αυτό σημαίνει ότι μπορούμε να δώσουμε ως όρισμα για πιο σωστά αποτελέσματα μια λίστα από μοναδικά tokens που εμφανίζονται στο έγγραφα.

Οι δημιουργοί του GSDMM έχουν παραθέσει ένα απλό παράδειγμα ώστε να γίνει κατανοητός ο τρόπος λειτουργίας του αλγορίθμου. Έχουν ονομάσει το παράδειγμα ως Movie Group Processκαι πάει ως εξής:

Έστω ότι ένα καθηγητής προσπαθεί να χωρίσει μια συλλογή μαθητών σε μικρότερα τραπέζια σε εστιατόριο βάση των ταινιών που βλέπουν ώστε να έχουν για κάτι να συζητήσουν. Αρχικά τους χωρίζει τυχαία σε K τραπέζια και τους βάζει να συμπληρώνουν μια μικρή λίστα από ταινίες που έχουν δει. Έπειτα ο κάθε μαθητης καλείτε να επιλέξει άλλο τραπέζι βάση των 2 κανόνων:

- Κανόνας 1: Επέλεξε το τραπέζι με περισσότερους μαθητές. Αυτός ο κανόνας βελτιώνει τη πληρότητα δηλαδή όλοι οι μαθητές που έχουν τα ίδια ενδιαφέροντα σε ταινίες να κάτσουν μαζί
- Κανόνας 2: Επέλεξε ένα τραπέζι που οι μαθητές έχουν παρόμοια ενδιαφέροντα σε ταινίες με εσένα. Αυτός ο κανόνας αυξάνει την ομοιογένεια δηλαδή στοχεύουμε στο όλοι οι μαθητές στο τραπέζι να έχουν τα ίδια ενδιαφέροντα σε ταινίες.

Αυτή η διαδικασία συνεχίζεται για όλους τους μαθητές με αποτέλεσμα κάποια τραπέζια να μεγαλώσουν και κάποια να μείνουν άδεια. Στο τέλος τα τραπέζια θα έχουν μαθητές με παρόμοια αν όχι ίδια ενδιαφέροντα σε ταινίες.

Στη βιβλιοθήκη sklearn δεν υπάρχει υλοποίηση του αλγορίθμου αλλά ο χρήστης rwalk (Ryan Walker) στο GitHub έχει δημιουργήσει μία για python. Επομένως με τις οδηγίες που προσφέρει μπορούμε να τον εκτελέσουμε αφού πρώτα δημιουργήσουμε μια λίστα από μοναδικά tokens όπως φράσεις δυο λέξεων. Αρχικά θα κάνουμε μια προ-επεξεργασία τα έγγραφα και μετά θα χρησιμοποιήσουμε το nltk.ngrams για τη δημιουργία της λίστας.

```
token_list = []
for i in normalized:
    bigrams = nltk.ngrams(i.split(), 2)
    w = []
    for grams in bigrams:
        #print(grams)
        w.append(grams)
    token_list.append(w)

from gsdmm import MovieGroupProcess

vocab = set(x for doc in token_list for x in doc)
n_terms = len(vocab)
print("Voc size:", n_terms)
print("Number of documents:", len(token_list))

mgp = MovieGroupProcess(K=10, alpha=0.1, beta=0.1, n_iters=40)

vocab = set(x for doc in docs for x in doc)
n_terms = len(vocab)
n_docs = len(docs)

# Fit the model on the data given the chosen seeds
y = mgp.fit(docs, n_terms)
```

```
Voc size: 8899
Number of documents: 314
In stage 0: transferred 235 clusters with 10 clusters populated
In stage 1: transferred 113 clusters with 10 clusters populated
In stage 2: transferred 71 clusters with 10 clusters populated
In stage 3: transferred 57 clusters with 10 clusters populated
In stage 4: transferred 58 clusters with 10 clusters populated
In stage 5: transferred 48 clusters with 10 clusters populated
```

Εικόνα 9: Εκτέλεση αλγορίθμου GSDMM

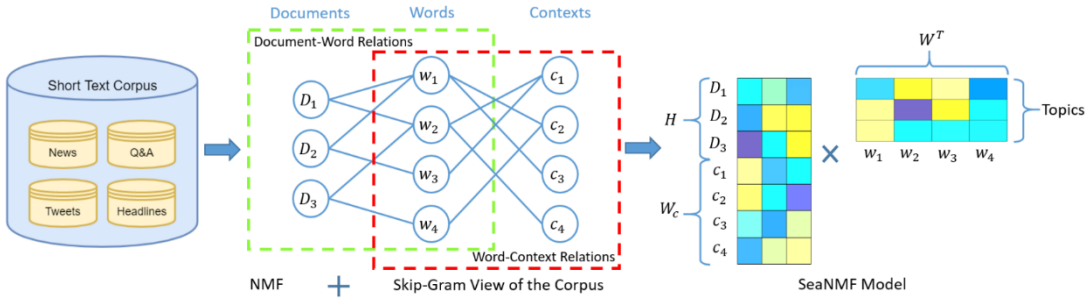
4.3.3 Semantics-assisted Non-negative Matrix Factorization (SeaNMF)

Ο αλγόριθμος Semantics-assisted Non-negative Matrix Factorization (SeaNMF) παρουσιάστηκε από τους Tian Shi, Kyeongpil Kang, Jaegul Choo and Chandan K. Reddy το 2018 ως μία βελτιωμένη εκδοχή του Non-negative Matrix Factorization (NMF), ειδικότερα στα μικρά κείμενα. Από την έρευνα τους παρατήρησαν ότι ο SeaNMF είναι μακράν καλύτερος του NMF και LDA για μικρά κείμενα όπως τα δεδομένα (datasets) Yahoo Answers, Tweets, GoogleNews και τίτλοι ομιλιών σε συνέδρια (Shi, 2018).

Σε μικρά κείμενα μία γνωστή λύση για το πρόβλημα εύρεσης θεμάτων είναι οι χρήση των σημασιολογικών (semantic) πληροφοριών για τις λέξεις από άλλες πήγες. Δηλαδή

ο αλγόριθμος θα βασίζετε σε ένα ήδη εκπαιδευμένο μοντέλο για τη παροχή πληροφοριών. Δυστυχώς, αυτό δεν είναι πάντα εφικτό διότι το μοντέλο χρειάζεται να εκπαιδευτεί με πολλά δεδομένα της ίδιας γλώσσας αλλά και του ίδιου τομέα με τα έγγραφα που εξετάζουμε.

Για αυτό το λόγο δημιουργήθηκε ο SeaNMF διότι χρησιμοποιεί ένα skip-gram μοντέλο με negative sampling (SGNS). Έχει αποδειχθεί ότι το SGNS μπορεί προσφέρει state-of-the-art επιδόσεις στην εύρεση συντακτικών και σημασιολογικών ομοιοτήτων σε λέξεις (Mikolov, 2013).. Αυτές οι συσχετίσεις θεωρούνται ως συνύπαρξη λέξεων και έτσι προσπαθούμε να λύσουμε το πρόβλημα έλλειψης δεδομένων.



Εικόνα 10: Επισκόπηση του αλγορίθμου SeaNMF (Shi, 2018)

Στη βιβλιοθήκη sklearn δεν υπάρχει υλοποίηση του αλγορίθμου αλλά ο δημιουργός του SeaNMF Tian Shi έχει δημιουργήσει μία για python στο GitHub. Επομένως με τις οδηγίες που προσφέρει μπορούμε να τον εκτελέσουμε με ευκολία διότι το μόνο που χρειάζεται είναι ένα αρχείο με προ-επεξεργασμένα έγγραφα.

```
with open('normalized_documents.txt', 'w', encoding='utf-8') as f:
    for line in normalized:
        print(line, file=f)
```

Εκτέλεση SeaNMF

Μόλις δημιουργήσουμε το αρχείο με τα προ-επεξεργασμένα έγγραφα, θα χρησιμοποιήσουμε αυτές τις εντολές στη γραμμή εντολών (CMD)

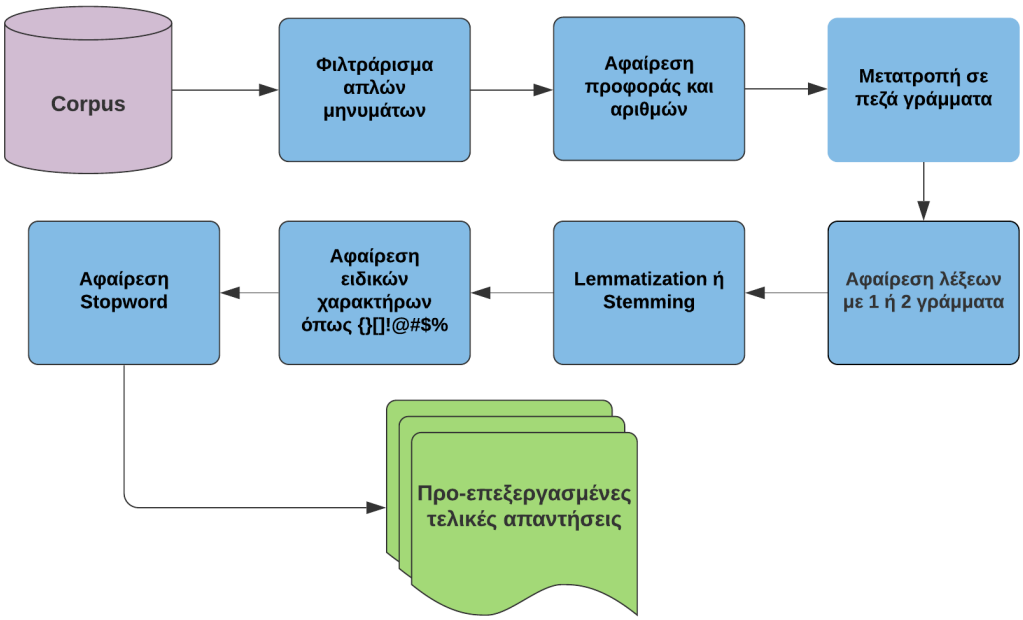
1. python3 data_process.py
2. python3 train.py --n_topics 10
3. python3 vis_topic.py

Εικόνα 11: Βήματα εκτέλεσης του αλγόριθμου SeaNMF

4.4 Προ επεξεργασια δεδομενων

Όταν τελειώσουμε με την εκκαθάριση των invalid session βάση της "Πολιτικής Εκκαθάρισης Συνόλου Δεδομένων" που αναπτύξαμε παραπάνω, θα έχουμε στα χεριά μας ένα σύνολο δεδομένων (dataset) από αξιολογά session που οι μαθητές έχουν παραθέσει απάντηση στο ερώτημα του MOOC. Παρατηρώντας προσεκτικά θα

προσέξουμε ότι έχουμε κάποια κενά ή μη χρήσιμα messages και κάποια σφάλματα με τα {player_id}. Αυτό όμως δεν μας επηρεάζει καθώς θα εφαρμόσουμε τους αλγορίθμους Μοντελοποίησης Θεμάτων μόνο στις τελικές απαντήσεις των μαθητών.



Εικόνα 12: Βήματα προ-επεξεργασίας των εγγράφων

4.4.1 Επεξεργασία των μηνυμάτων

Για να εκτελέσουμε τους αλγορίθμους χρειάζεται να κάνουμε μια προ-επεξεργασία στα δεδομένα μας. Σε αυτό το στάδια θα γίνουν κάποια απαραίτητα βήματα όπως το φιλτράρισμα των απλών μηνυμάτων αλλά και η δημιουργία ενός document-word matrix που θα είναι το όρισμα στους αλγορίθμους μας. Επίσης προσθέσαμε περισσότερα βήματα για την μείωση του λεξιλογίου, αφαίρεση χαρακτήρων και stopword που θα επηρεάσουν σημαντικά τα αποτελέσματα μας.

4.4.2 Φιλτράρισμα μηνυμάτων

Από τα καθαρισμένα δεδομένα που περιέχουν μόνο valid session θα χρειαστούμε μόνο τα μηνύματα με message_type = 0 δηλαδή μόνο τις τελικές απαντήσεις και με length>0.

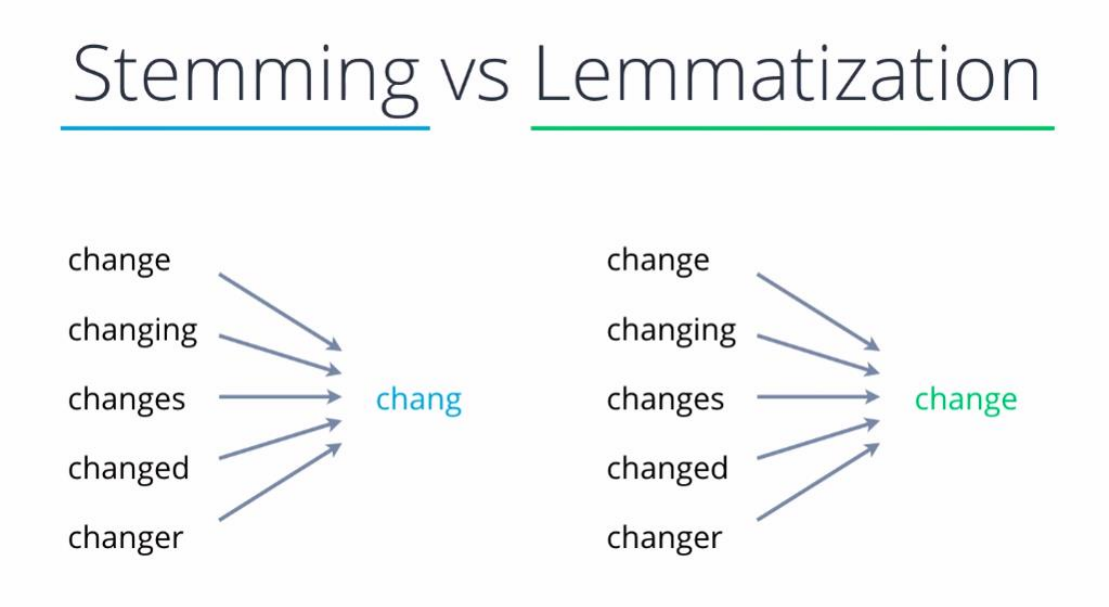
```
mooc = pd.read_excel(m_file,
    header=0,
    index_col=False,
    keep_default_na=True
    )

    for index, row in mooc.iterrows():
        if(row['message_type'] == 0 and int(row['message_length']) > 0):
            answers.append(row['the_message'])
```

```
Final answers found: 314
Total Messages: 17511
```

Εικόνα 13: Κώδικας φιλτραρίσματος μηνυμάτων και αποτελέσματα

4.4.3 Lemmatization

Το Lemmatization είναι μια διεργασία μετατροπής μιας λέξης στη βασική της μορφή, δηλαδή θα υλοποιήσει περικοπή των κλιτικών καταλήξεων και αναγωγή παράγωγων μορφών μιας λέξης σε κοινή βασική μορφή. Η διαφορά μεταξύ του Lemmatization και Stemming είναι, το Lemmatization εξετάζει το πλαίσιο και μετατρέπει τη λέξη σε μια ουσιαστική μορφή , ενώ το Stemming απλώς αφαιρεί τους τελευταίους χαρακτήρες, οδηγώντας συχνά σε λανθασμένες έννοιες και ορθογραφικά λάθη. Το Lemmatization αποτελεί εργασία δυσκολότερη και πιο χρονοβόρα σε σύγκριση με το Stemming διότι έχει το εξτρά βήμα που για να αφαιρέσει το επίθεμα πρέπει να βρει τη λέξη στο λεξικό.



Εικόνα 14: Διαφορές Stemming και Lemmatization (LaptrinhX, 2020)

‘Caring’->Lemmatization->‘Care’
‘Caring’ -> Stemming -> ‘Car’

Χρησιμοποιούμε Lemmatization ή Stemming ώστε να μειώσουμε το πλήθος των μοναδικών λέξεων που έχουμε στο λεξικό μας. Αυτό έχει ως αποτέλεσμα να έχουμε μικρότερο document-word matrix όταν θα το χρειαστούμε παρακάτω και ο αλγόριθμος topic modeling θα μπορεί να βγάλει καλύτερα αποτελέσματα.

Στην εργασία θα χρησιμοποιήσουμε Lemmatization επειδή διατηρεί πραγματικές λέξεις και δεν τις κόβει όπως το Stemming. Επίσης η βιβλιοθήκη Spacy προσφέρει έναν από τους καλύτερους Lemmatizers για την Ελληνική γλώσσα.

```
# How to install spacy greek lemmatizer
# conda install -c conda-forge spacy
# python -m spacy download el_core_news_lg
import spacy
import el_core_news_lg

original_text = "θα χρησιμοποιήσουμε λεξικό ,καθώς είναι ταχύτερη και απλούστερη μέθοδος απο την λίστα. Ο χρήστης εισάγει το κλει
print("**Αρχικό κείμενο**")
print(original_text)
print("_______")
nlp = el_core_news_lg.load()
text = nlp(original_text)
text = ' '.join([word.lemma_ if word.lemma_ != '-PRON-' else word.text for word in text])
print("**Lemmatized κείμενο**")
print(text)
```

```
**Αρχικό κείμενο**
θα χρησιμοποιήσουμε λεξικό ,καθώς είναι ταχύτερη και απλούστερη μέθοδος απο την λίστα. Ο χρήστης εισάγει το κλειδί , και εμφανί
ζεται ή αντίστοιχη τιμή . Όσα ζεύγη και να εχω , ο χρόνος εντοπισμού παραμένει ίδιος.

_______
**Lemmatized κείμενο**
θα χρησιμοποιήσω λεξικό , καθώς είναι ταχός και απλούστερη μέθοδος απο την λίστα . ο χρήστης εισάγω το κλειδί , και εμφανίζομαι
ή αντίστοιχη τιμή . όσα ζεύγη και να εχω , ο χρόνος εντοπισμός παραμένω ίδιος .
```

Εικόνα 15: Εφαρμογή Spacy Lemmatizer

4.4.4 Stopwords

Τα stopwords είναι λέξεις που εμφανίζονται συχνά στη γλώσσα αλλά συμβάλλουν ελάχιστα στο γενικό νόημα του εγγράφου. Λέξεις όπως το "να", "μια", "είναι", "υστέρα", "όπως" κ.λπ. δεν συμβάλλουν στο νόημα. Μπορούμε να τις αφαιρέσουμε ώστε να γίνει γρηγορότερα η επεξεργασία του εγγράφου αλλά και για να παράγουμε καλύτερα αποτελέσματα με τους αλγορίθμους που θα εφαρμόσουμε.

Στη συγκεκριμένη εργασία ξεκινήσαμε να χρησιμοποιούμε μια λίστα από 800 ελληνικά stopwords. Μετά από ερευνά παρατηρήσαμε πως υπάρχουν παραπάνω λέξεις που δεν επηρεάζουν το νόημα διότι είναι σχεδόν σε κάθε έγγραφο. Κάποια από τα νέα stopwords είναι: {“νίκο”, “κατερίνα”, “λίστα”, “nikou”, “katerina”, “ελληνικη”, “λεξη”, “ελληνικες”, “λεξει”, “λυση”}. Αρά είδαμε πως κάποιος μπορεί να ξεκινήσει με μια επιμελημένη λίστα από stopwords και έπειτα την ενημερώνει ανάλογα τα δεδομένα του.

Όταν έχουμε τη λίστα μας έτοιμη μπορούμε να αρχίσουμε το φιλτράρισμα. Αρχικά σπάσαμε το κάθε έγγραφο σε tokens με τον TokTokTokenizer της βιβλιοθήκης nltk διότι είναι γρήγορος και το μόνο που χρειαζόμασταν ήταν να βρίσκει τις λέξεις χωρίς κάποιες εξιδεικευμένες συνθήκες. Έπειτα περάσαμε όλα τα tokens από τη λίστα με stopwords και κρατήσαμε μόνο εκείνα που δεν υπήρχαν στη λίστα μας. Τέλος ενώσαμε όλα τα φιλτραρισμένα tokens μετάξι τους με ένα κενό ανάμεσα τους.

```
from nltk.tokenize.toktok import ToktokTokenizer
tokenizer = ToktokTokenizer()

original_text = "θα χρησιμοποιήσουμε λεξικό ,καθώς είναι ταχύτερη και απλούστερη μέθοδος απο την λίστα. Ο χρήστης εισάγει το κλε
print("**Αρχικό κείμενο**")
print(original_text)
tokens = tokenizer.tokenize(original_text)
tokens = [token.strip() for token in tokens]
filtered_tokens = [token for token in tokens if token not in stopwords]
filtered_text = ' '.join(filtered_tokens)
print("\n**Κείμενο χωρίς stopwords**")
print(filtered_text)
```

```
**Αρχικό κείμενο**
θα χρησιμοποιήσουμε λεξικό ,καθώς είναι ταχύτερη και απλούστερη μέθοδος απο την λίστα. Ο χρήστης εισάγει το κλειδί , και εμφανί
ζεται ή αντίστοιχη τιμή . Όσα ζεύγη και να εχω , ο χρόνος εντοπισμού παραμένει ίδιος.

**Κείμενο χωρίς stopwords**
χρησιμοποιήσουμε λεξικό , καθώς ταχύτερη απλούστερη μέθοδος λίστα. Ο χρήστης εισάγει κλειδί , εμφανίζεται ή αντίστοιχη τιμή . Ό
σα ζεύγη , χρόνος εντοπισμού παραμένει ίδιος .
```

Εικόνα 16: Αφαίρεση stopword

4.4.5 Μοντέλο Bag-of-Words

Αφού τελειώσουμε με όλα τα παραπάνω βήματα προ-επεξεργασίας θα μεταβούμε στο τελικό βήματα που είναι η μετατροπή των επεξεργασμένων εγγράφων σε ένα bag-of-words μοντέλο. Το μοντέλο “τσάντας λέξεων” (bag-of-words) είναι στην ουσία μια αναπαράσταση των εγγράφων ως ένας πίνακας λέξεων που η γραμμές είναι τα έγγραφα μας και οι στήλες όλα τα μοναδικά tokens. Τα tokens μπορεί να είναι λέξεις, φράσεις, αριθμοί και άλλα σύμβολα. Κυρίο βήμα είναι το Tokenization που χωρίζει τα μεγαλύτερα κομμάτια του κειμένου σε προτάσεις → φράσεις → λέξεις κ.λπ. Στο τέλος θα μείνουμε με ένα πίνακα που στην απλή μορφή του κάθε κελί δηλώνει τη συχνότητα του όρου (term) X στο έγγραφο Y, σημειώνεται με 0 και 1+ ή με σταθμισμένες τιμές όπως θα δούμε παρακάτω.

Το bag-of-words μοντέλο πήρε αυτό το όνομα διότι στο τέλος κάθε έγγραφο αναπαριστάτε από "μια τσάντα από λέξεις", χωρίς κάποια γραμματική ή ταξινόμηση. Είναι μία απλή τεχνική ανάλυσης που μπορούμε να την παραμετροποιήσουμε ώστε να αφαιρέσουμε το θόρυβο (δηλαδή τις σπάνιες λέξεις) και να μειώσουμε τις διαστάσεις του πίνακα.

Κάποια από τα μειονεκτήματα αυτής της μεθόδου:

1. Νόημα: αφαιρώντας στη σύνταξη της πρότασης είναι πολύ πιθανό να χάσουμε και το νόημα της. Για παράδειγμα το "αυτό είναι αποδοτικό" με το "είναι αυτό αποδοτικό" ή η λέξη "πολύ" από μόνη της δεν βοηθάει.
2. Λεξικό: το λεξικό χρειάζεται προσοχή διότι το μεγάλο λεξιλόγιο θα αυξήσει το μέγεθος και την χρονική πολυπλοκότητα. Επίσης αν κρατάμε τις σπάνιες λέξεις ο τελικός πίνακας θα γεμίζει μηδενικά που επιβαρύνουν περισσότερο το σύστημα και δεν βελτιώνουν τα τελικά αποτελέσματα.

	Λέξη 1	Λέξη 2	Λέξη 3	Λέξη 4	Λέξη 5
Έγγραφο 1	1	1	0	1	0

Έγγραφο 2	0	0	1	1	0
Έγγραφο 3	0	1	1	1	1

Δηλαδή:

Έγγραφο 1 = [1, 1, 0, 1, 0]

Έγγραφο 2 = [0, 0, 1, 1, 0]

Έγγραφο 3 = [0, 1, 1, 1, 1]

	Λέξη 1	Λέξη 2	Λέξη 3	Λέξη 4	Λέξη 5
Έγγραφο 1	0.72	0.55	0.00	0.43	0.00
Έγγραφο 2	0.00	0.00	0.79	0.61	0.00
Έγγραφο 3	0.00	0.48	0.48	0.37	0.63

Δηλαδή:

Έγγραφο 1 = [0.72, 0.55, 0.00, 0.43, 0.00]

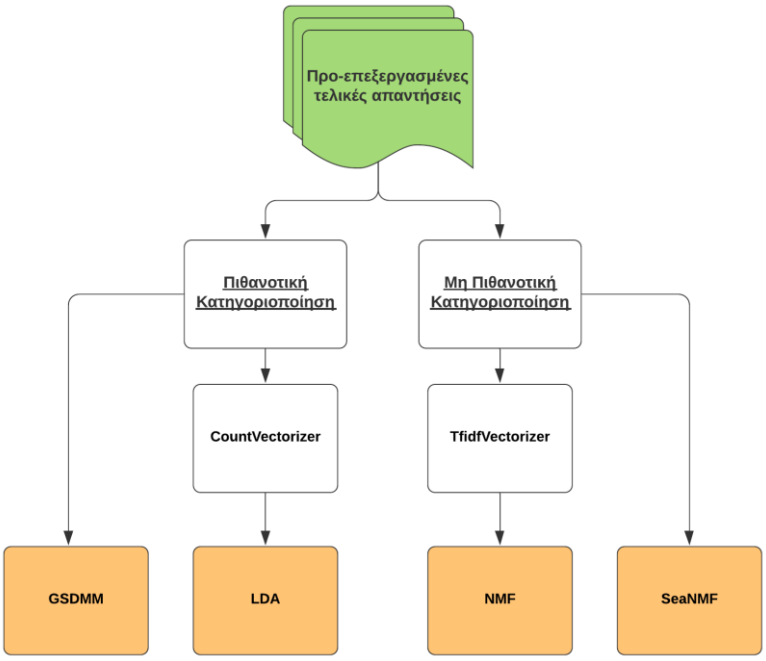
Έγγραφο 2 = [0.00, 0.00, 0.79, 0.61, 0.00]

Έγγραφο 3 = [0.00, 0.48, 0.48, 0.37, 0.63]

4.4.6 Δημιουργία συνόλου δεδομένων προς ανάλυση

Για τους αλγόριθμους που θα εξετάσουμε, θα χρειαστούμε δυο διαφορετικά μοντέλα bag-of-words ως ορίσματα.

- Το CountVectorizer (word count) για LDA, GSDMM και SeaNMF
- Το TfidfVectorizer (TF-IDF) για NMF



Εικόνα 17: Δημιουργία συνόλου δεδομένων προς ανάλυση

Αρχίζοντας με τον CountVectorizer που είναι ο απλούστερος από τους δυο, μας δίνει τη πιο βασική αριθμητική αναπαράσταση των εγγράφων με διανύσματα καταμέτρησης. Δηλαδή δημιουργεί διάνυσμα για κάθε έγγραφο που έχει μέγεθος ίσο με το μέγεθος του λεξιλογίου. Έστω ότι ο πίνακας είναι γεμάτος μηδενικά, ο CountVectorizer θα περνούσε κάθε έγγραφο και θα αύξανε τη τιμή κατά 1 αν έβλεπε εκείνη τον όρο μέσα στο έγγραφο και θα αφήσει 0 οπού δεν τον βλέπει. Επειδή αυτά τα διανύσματα έχουν πολλά μηδενικά τα ονομάζουμε αραιά διανύσματα (sparse vectors).

```
document = ["και ο καιρός και σήμερα είναι καλός",
            "και αυτός ο αυτοκινητόδρομος είναι καλός",
            "οδηγάμε προσεκτικά όταν ο καιρός δεν είναι βροχερός"]
```

	αυτοκινητόδρομος	αυτός	βροχερός	δεν	και	καιρός	καλός	οδηγάμε	προσεκτικά	σήμερα	όταν
0	0	0	0	0	2	1	1	0	0	1	0
1	1	1	0	0	1	0	1	0	0	0	0
2	0	0	1	1	0	1	0	1	1	0	1

Εικόνα 18: Αποτέλεσμα CountVectorizer από ένα δείγμα εγγράφων

Όπως μπορούμε να δούμε ο CountVectorizer αν και εύκολος στη κατανόηση είναι πολύ απλοϊκός. Το βασικό του πρόβλημα είναι ότι οι κοινές και οι σπάνιες λέξεις έχουν την ίδια αξία στο διάνυσμα. Για παράδειγμα η λέξη "το" σε σχέση με τη λέξη "πτήση" είναι λογικό να εμφανίζεται πιο συχνά αλλά δεν επηρεάζει καθόλου το νόημα.

Μια λύση στο πρόβλημα είναι αντί να μετράμε το για κάθε όρο αν εμφανίστηκε με >=1 και αν δεν εμφανίστηκε με 0, είναι να σταθμίσουμε τις τιμές. Αυτό το επιτυγχάνουμε

με το να υπολογίζουμε τη συχνότητα εμφάνισης του όρου σε όλα τα έγγραφα, έτσι ώστε όροι όπως το "το" να έχει μικρή τιμή.

Η τεχνική ονομάζεται Term Frequency – Inverse Document Frequency ή TF-IDF που απαρτίζεται από 2 τιμές-σκορ:

- Term Frequency: Είναι μια τιμή που συνοψίζει τη συχνότητα εμφάνισης της λέξης στο συγκεκριμένο έγγραφο.
- Inverse Document Frequency: Είναι μια τιμή που δείχνει πόσο σπάνια είναι η λέξη μέσα σε όλα τα έγγραφα. Αρά μια πολύ συχνή λέξη θα έχει μικρή τιμή.

$$w(t,D) = tf(t,d) \ * (\log\left(\frac{n}{df(t)}\right) + 1)$$

Όπου n = ο αριθμός των εγγράφων.

Για παράδειγμα η λέξη αυτοκίνητο εμφανίζεται στο συγκεκριμένο έγγραφο 3 φορές και εμφανίζεται 50 φορές γενικά σε όλο το corpus των 1000 εγγράφων

$$w(t,D) = 3 \ * \left(\log\left(\frac{1000}{50}\right) + 1\right) = 11{,}987$$

```
document = ["και ο καιρός και σήμερα είναι καλός",
            "και αυτός ο αυτοκινητόδρομος είναι καλός",
            "οδηγάμε προσεκτικά όταν ο καιρός δεν είναι βροχερός"]
```

	αυτοκινητόδρομος	αυτός	βροχερός	δεν	είναι	και	καιρός	καλός	οδηγάμε	προσεκτικά	σήμερα	όταν
0	0.000000	0.000000	0.000000	0.000000	0.269040	0.692876	0.346438	0.346438	0.000000	0.000000	0.455524	0.000000
1	0.534093	0.534093	0.000000	0.000000	0.315444	0.406192	0.000000	0.406192	0.000000	0.000000	0.000000	0.000000
2	0.000000	0.000000	0.410747	0.410747	0.242594	0.000000	0.312384	0.000000	0.410747	0.410747	0.000000	0.410747

Εικόνα 19: Αποτέλεσμα TfidfVectorizer από ένα δείγμα εγγράφων

Δηλαδή αν μια λέξη που εμφανίζεται πολλές φορές σε ένα έγγραφο και σχετικά λίγες σε όλα τα άλλα έγγραφα θα έχει υψηλότερη τιμή σε σχέση με άλλη λέξη που εμφανίζεται σε κάθε έγγραφο ή εμφανίζεται πολύ σπάνια. Ως αποτέλεσμα ο αλγόριθμος τονίζει της λέξεις που παρέχουν χρήσιμη πληροφορία σε κάθε έγγραφο.

Το CountVectorizer και το TfidfVectorizer έχουν υλοποιηθεί στη βιβλιοθήκη sklearn που θα χρησιμοποιήσουμε. Πριν κάνουμε τη τελική εξέταση θα χρειαστεί να δοκιμάσουμε διάφορες παραμέτρους για ένα καλύτερο bag-of-words μοντέλο. Οι κυρίες παράμετροι είναι:

- min_df: ο όρος να εμφανίζεται τουλάχιστον σε x έγγραφα (πχ min_df = 10) ή τουλάχιστον στο p.% του corpus (πχ min_df = .02 δηλαδή στο 2% του corpus)
- max_df: ο όρος να εμφανίζεται το πολύ σε x έγγραφα (πχ max_df= 1000) ή το πολύ στο p% του corpus (πχ max_df= .95 δηλαδή στο 95% του corpus)
- max_features: ο μέγιστος αριθμός όρων που θα έχει το μοντέλο. Αν δεν είναι (max_features = None) τότε παίρνει τους a (a = max_features) τοπ όρους βάση της συχνότητας εμφάνισης στο corpus

- ngram_range: το μέγεθός των όρων μας πχ ngram_range=(1,1) παίρνει μόνο λέξεις, με (2,2) περιέχει μόνο φράσεις των 2 λέξεων, με (1,3) έχει λέξεις και φράσεις μέχρι 3 λέξεις

Η επιλογή του κατάλληλου ngram_range επηρεάζει σημαντικά το αποτέλεσμα του topic modeling. Από τις προηγούμενες εξετάσεις που κάναμε παρατηρήσαμε πώς με φράσεις των 2 λέξεων ο αλγόριθμος Latent Dirichlet Allocation (LDA) απέδωσε πολύ καλύτερα. Για αυτό το λόγο θα χρησιμοποιήσουμε ngram_range = (2,2) για όλους τους αλγόριθμους ώστε αυτή η παράμετρος να μην επηρεάσει τη σύγκριση μεταξύ αλγορίθμων.

```
from sklearn.feature_extraction.text import CountVectorizer

count_vectorizer = CountVectorizer(min_df=0., max_df=.95,ngram_range=(1, 1))
bag_of_words = count_vectorizer.fit_transform(normalized)
bag_of_words = bag_of_words.toarray()
vocab = count_vectorizer.get_feature_names()
pd.DataFrame(bag_of_words, columns=vocab)
```

	accurate	adict	agent	agglikes	alias	alist	apple	ascci	ascii	associate	...	χωριστω	χωρο	ψαξει	ψαξω	ψαρι	ψαχνω	ψηφιζω	ωραιοποιω	ωραι
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	1	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
309	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
310	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
311	0	0	0	0	0	0	0	0	0	0	...	0	0	1	0	0	0	0	0	
312	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
313	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	

314 rows × 2089 columns

Εικόνα 20: Κώδικα και εκτέλεση CountVectorizer για unigrams (ngrams = 1)

```
from sklearn.feature_extraction.text import CountVectorizer

count_vectorizer = CountVectorizer(min_df=0., max_df=.95,ngram_range=(2, 2))
bag_of_words = count_vectorizer.fit_transform(normalized)
bag_of_words = bag_of_words.toarray()
vocab = count_vectorizer.get_feature_names()
pd.DataFrame(bag_of_words, columns=vocab)
```

	accurate αποτελεσματας	adict alias	adict alist	adict else	adict gr_list	adict key_list	adict print	adict word	adict αγγλικα	adict αγγλικη	...	ψαχνω τελευταιος	ψαχνω τιμη	ψαχνω τροπο	ωραιοποιω εξοδο	ωραιος ομορφο	ωστοσος δημιουργηθω	
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
309	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
310	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
311	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
312	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
313	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	

314 rows × 8936 columns

Εικόνα 21: Κώδικα και εκτέλεση CountVectorizer για bigrams (ngrams = 2)

```
from sklearn.feature_extraction.text import TfidfVectorizer
n_features = 1000

tfidf_vectorizer = TfidfVectorizer(min_df=2,
                                   max_df=0.95,
                                   max_features=n_features,
                                   ngram_range=(2, 2)
                                  )
tfidf = tfidf_vectorizer.fit_transform(normalized)
tfidf = tfidf.toarray()
vocab = tfidf_vectorizer.get_feature_names()
pd.DataFrame(tfidf, columns=vocab)
```

	adict word	dict comprehension	for range	for αγγλικης	hash table	identifier συνεχειας	input δωσε	key value	langdict αγγλικη	mapping structure	...	χρονο προσπελασω	χρονο χρειαζομαι	χρονο χρειαζομαστε	χρονοβορο γιατι	) εντο
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
309	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
310	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
311	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
312	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
313	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	

314 rows × 1000 columns

Εικόνα 22: Κώδικα και εκτέλεση TfidfVectorizer για bigrams (ngrams = 2)

Όπως βλέπουμε ο TF-IDF μας βγάζει πιο στοχευμένα αποτελέσματα για τη κάθε λέξη. Εμείς όμως επιλέξαμε να χρησιμοποιήσουμε τον απλό CountVectorizer για τους αλγορίθμους LDA και GSDMM (παραλλαγή του LDA). Αυτό έγινε διότι όπως αναφέρει ο Blei (δημιουργός του LDA) στο paper του με τίτλο "Latent Dirichlet Allocation", ο LDA αντιμετωπίζει της αδυναμίες του TF-IDF και δεν χρησιμοποιεί την τεχνική (Blei, 2003). Το TF-IDF χρησιμοποιείτε για αλγεβρικά μοντέλα όπως το LSA και το NMF ενώ τα πιθανολογικά μοντέλα όπως το LDA προσπαθούν να υπολογίσουν τις πιθανότητες των κατανομών έγγραφο-θέμα και θέμα-λέξη δεν χρειάζονται σταθμισμένες τιμές. Αν χρησιμοποιούσαμε TF-IDF σε πιθανολογικό μοντέλο θα επηρέαζε το αποτέλεσμα διότι οι τιμές των σπάνιων όρων μπορεί να φτάσουν τις τιμές κάποιων κοινών όρων.

4.5 ΑΠΟΤΕΛΕΣΜΑΤΑ

Στη συγκεκριμένη εργασία θα εξετάσουμε τους αλγόριθμους LDA, NMF, GSDMM και SeaNMF. Με αυτή τη μίξη ορισμένων γνωστών και κάποιων εξειδικευμένων αλγόριθμων θα μπορέσουμε να έχουμε μια σφαιρική εικόνα για το αποτέλεσμα.

Οι αλγόριθμοι LDA και NMF είναι εύκολα προσβάσιμοι επειδή υπάρχει η υλοποίηση τους στη γνωστή βιβλιοθήκη sklearn. Για τους αλγόριθμους GSDMM και SeaNMF χρειάστηκε να βρούμε την υλοποίηση τους στο διαδίκτυο όπως αναφέραμε παραπάνω. Επιπλέον για τους GSDMM και SeaNMF κάναμε κάποιες αλλαγές για να υποστηρίζουν UTF-8 κωδικοποίηση καθώς δουλεύουμε με ελληνικούς χαρακτήρες.

Θα εξετάσουμε κάθε αλγόριθμο και στα δυο σύνολα δεδομένων (datasets) (πληθυσμός SPOC και πληθυσμος MOOC) και με εναλλαγή του αριθμού θεμάτων από 10 σε 5.

Τα δύο σύνολα δεδομένων που θα ερευνήσουμε έχουν απαντήσεις σε αλγοριθμικά ερωτήματα και οι μαθητές στις απαντήσεις τους εξηγούσαν τη βέλτιστη λύση. Επομένως είναι λογικό να περιμένουμε γενικότερες θεματικές όπως πολυπλοκότητα (χωρική και χρονική), λίστες και τιμές αλλά και υλοποίηση του αλγορίθμου.

Τέλος, όλοι οι αλγόριθμοι θα έχουν ως όρισμα ένα bag-of-words μοντέλο από φράσεις 2 λέξεων (bigrams). Οι αλγόριθμοι GSDMM και SeaNMF υλοποιούν μόνοι τους ένα απλοϊκό bag-of-words και για αυτό, βρίσκουμε από πριν τα bigrams και τα ενώνουμε για να κάνουμε μία αναπαράσταση του εγγράφου.

```
docs = []
for i in normalized:
    bigrams = nltk.ngrams(i.split(), 2)
    w = []
    for grams in bigrams:
        string_bigram = '_'.join(grams)
        w.append(string_bigram)
    doc = ' '.join(w)
    docs.append(doc)
print(docs)
```

```
['καλυτερη_γνωμη γνωμη_χρηση χρηση_λιστας λιστας_γινω γινω_χρηση χρηση
νιζω εμφανιζω_στοιχειο στοιχειο_υπαρχω υπαρχω_μεταφρασμενη μεταφρασμεν
κατω_κωδικα κωδικα_word word_input input_δωστε δωστε_επιθυμειτε επιθυμ
κυλοςdogγατακατ_print print_adict adict_word word_κυριο κυριο_πλεονεκτ
ς γρηγορος_ταχυτητο ταχυτητο_κωδικο κωδικο_επηρεαζομαι επηρεαζομαι_πλη
```

Εικόνα 23: Δημιουργία εγγράφων μόνο από bigrams

4.5.1 Πληθυσμός SPOC με 10 θέματα

Θέμα	Όροι
1	τιμες λιστε, χωρο μνημη, αντιστοιχες αγγλικες, ελληνικης λεξης, χρονο αυξανομαι
2	χρονο εκτελεση, κλειδιος τιμες, εκτελεση προγραμμα, χρηση λεξικων, κλειδια τιμες
3	χρονο εκτελεση, δυο λιστε, αντιστοιχη αγγλικη, αντιστοιχες αγγλικα, συνδυασμο λιστας
4	δομη δεδομενο, μνημη υπολογιστη, λιγοτερη μνημη, δυο λιστες, πληθο στοιχεια
5	χρονο προσπελαση, ταχυτητο εκτελεση, αγγλικων λεξεο, λιστε τροπο, κυριο πλεονεκτημας
6	χωρο μνημη, ελληνικης γλωσσα, μπορουσε χρηση, χρονο προσπελαση, λιστε προτεινω

Θέμα	Όροι
7	αντιστοιχη τιμη, αντιστοιχες αγγλικες, τιμη κλειδιο, ιδιος θεση, αντιστοιχες αγγλικα
8	αντιστοιχη αγγλικη, χρονο εκτελεση, λιγοτερος χρονο, ελληνικης γλωσσα, χρησιμοποιω κλειδι
9	αντιστοιχη αγγλικη, κλειδι μπορω, βρω κλειδι, ευρεση αγγλικης, θεση στοιχειου
10	αντιστοιχες αγγλικες, χρονο εντοπισμο, χρηση λιστας, περισσοτερος χρονο, χρειαζομαι υπολογιστης

Πίνακας 5: LDA 10 θέματα για τελικές απαντήσεις SPOC

Θέμα	Όροι
1	αντιστοιχες αγγλικες, τιμες αντιστοιχες, αντιστοιχη τιμη, κλειδι τιμες, χρηστης πληκτρολογω
2	χωρο μνημη, λιγοτερος χρονο, εκτελεση κωδικο, λιγοτερος χωρο, κατανοηση κωδικο
3	αντιστοιχη αγγλικη, ελληνικης γλωσσα, ελληνικε αγγλικες, αντιστοιχες αγγλικης, προγραμμας γρηγορο
4	συνδυασμο λιστας, δυο λιστε, λιστε προτεινω, υπαρχω κλειδι, κωδικα γινω
5	περισσοτερος χρονο, αυξανομαι χρονο, χρηση δυο, προσπελαση δεδομενος, ελληνικης λεξης
6	λιγοτερη μνημη, πληθο λεξεο, περιεχει λεξικα, αυξανομαι πληθο, κυριο πλεονεκτημας
7	εκτελεση προγραμμα, αντιστοιχη αγγλικα, καταλληλη βιβλιοθηκη, εντολη ascci, βρω κλειδι
8	μεταφρασει αγγλικα, αντιστοιχες αγγλικα, χρονο εκτελεση, κλειδι τιμη, εκτελεση μικροτερος
9	καλυτερος τροπο, χρονο προσπελαση, κλειδι αντιστοιχω, τιμες λιστε, αντιστοιχη γλωσσας
10	τιμες κλειδια, ευρεση αγγλικης, χρονο χρειαζομαι, τιμη κλειδιο, χρονο απαιτηται

Πίνακας 6: NMF 10 θέματα για τελικές απαντήσεις SPOC

Θέμα	Όροι
1	αντιστοιχη_αγγλικη, χρηση_λεξικων, χρονο_εκτελεση, μπορω_κατευθειαν, κατευθειαν_βρω
2	κλειδια_τιμες, χρονο_χρειαζομαι, ευρεση_αγγλικης, κυριο_πλεονεκτημας, αγγλικων_λεξεο
3	χρονο_εκτελεση, αντιστοιχες_αγγλικα, κλειδια_αντιστοιχες, αντιστοιχες_αγγλικες, τιμη_κλειδιο
4	τιμες_αντιστοιχες, αντιστοιχες_αγγλικες, αντιστοιχη_αγγλικη, κλειδιος_τιμες, υπαρχω_κλειδι
5	τιμες_λιστε, δομη_δεδομενο, δεδομενο_λεξικα, αγγλικα_πλεονεκτημας, συνωνυμος_λεξεο
6	χρονο_εντοπισμο, αυξανομαι_χρονο, αντιστοιχη_τιμη, δεδομενο_λιστος, χρονο_εντοπισουμε
7	χρονο_προσπελαση, ιδιος_ανεξαρτητα, χρηση_κλειδι, γινει_προσπελαση, ελληνικης_λεξης
8	χωρο_μνημη, λιγοτερος_χρονο, λιγοτερος_χωρο, μνημη_υπολογιστη, χρηση_dict
9	ελληνικης_γλωσσα, αντιστοιχη_αγγλικη, αντιστοιχη_μεταφρασμενη, μεταφρασμενη_αγγλικα, χρονο_εκτελεση
10	αντιστοιχη_τιμη, συνθετης_δομης, τελω_καταληγω, αντιστοιχες_αγγλικες, δεδομενο_συγκεντρωμενα

Πίνακας 7: GSDMM 10 θέματα για τελικές απαντήσεις SPOC

Θέμα	Όροι
1	αντιστοιχες_αγγλικα, τιμες_λιστε, μεταφρασει_αγγλικα, συνδυασμο_λιστας, τροπο_νικου
2	αντιστοιχη_τιμη, εδωσε_χρηστης, τιμη_κλειδιο, αντιστοιχες_αγγλικες, τιμες_αντιστοιχες
3	χρονο_εκτελεση, χρηση_λεξικων, λιγοτερη_μνημη, νικου_απαιτω, εκτελεση_προγραμμα
4	αντιστοιχες_αγγλικες, τιμες_αντιστοιχες, κλειδιος_τιμες, συνδυασμο_λιστας, χρηστης_πληκτρολογω
5	αντιστοιχη_αγγλικη, ελληνικης_γλωσσα, λιγοτερος_χρονο, εδωσε_χρηστης, χρονο_εκτελεση
6	χωρο_μνημη, λιγοτερος_χρονο, μνημη_υπολογιστη, ευρεση_αγγλικης, εκτελεση_κωδικο

Θέμα	Όροι
7	εκτελεση_προγραμμα, κυριο_πλεονεκτημας, δυο_λιστε, τιμες_αντιστοιχη, συνθετης_δομης
8	δομη_δεδομενο, συνθετη_δομη, τιμες_λιστε, συνθετης_δομης, χρηση_δυο
9	χρονο_εντοπισμο, αυξανομαι_χρονο, λεξικα_λιστε, χρηση_λιστας, πιστευω_δημιουργιας
10	χρονο_χρειαζομαι, κλειδια_τιμες, κλειδι_μπορω, περισσοτερος_χρονο, ιδιος_ανεξαρτητα

Πίνακας 8: SeaNMF 10 θέματα για τελικές απαντήσεις SPOC

Παρατηρήσεις:

- LDA: Διακρίνουμε 2 ομάδες θεμάτων η μία μιλάει για αγγλικές και ελληνικές τιμές ενώ η άλλη ομάδα αναφέρεται στο χρόνο εκτέλεσης και τη μνήμη, αλλά σχεδόν κανένα από τα θέματα δεν ανήκει αποκλειστικά σε μια από τις δύο ομάδες.
- NMF: εδώ μπορούμε να παρατηρήσουμε καλύτερα τα ξεχωριστά θέματα όπως το θέμα 2 που αφορά την πολυπλοκότητα ή το θέμα 1 που αφορά τιμές κλειδιά
- GSDMM: μόνο το θέμα 8 είναι τελείως ξεκάθαρο διότι όλες οι λέξεις του αφορούν το ίδιο πράγμα
- SeaNMF: τα περισσότερα θέμα έχουν μια μίξη λέξεων που αφορούν διαφορετικά πράγματα. Από την άλλη μπορούμε να παρατηρήσουμε στα τελευταία θέματα (7, 8 και 9) εμφανίζεται μια νέα ομάδα θεμάτων που αφορά την υλοποίηση της λύσης.

4.5.2 Πληθυσμός MOOC με 10 θέματα

Θέμα	Όροι
1	αγγλικης γλωσσα, μειονεκτημο μπορω, ελληνικα αγγλικα, κλειδι υπαρχω, αποδοτικο γρηγορος
2	συνθετη δομη, χρονο αναζητηση, κλειδι τιμη, βασικος πλεονεκτημας, δομη δεδομενο
3	σταθερος ανεξαρτη, ανεξαρτη πληθο, χρονος προσδιορισμου, προσδιορισμου τιμης, τιμης σταθερος
4	προσπελαση δεδομενος, αντιστοιχη αγγλικα, στοιχειο λιστας, πλεονεκτημας χρηση, λιστε λεξικα
5	κλειδι τιμη, αγγλικη γλωσσας, δυο λιστε, ελληνικης γλωσσα, αντιστοιχη αγγλικα
6	κλειδι τιμη, langdict αγγλικη, δομη συνδυαζω, μειονεκτημο μπορω, μεταλλαξιμη δομη
7	μεταφραση αγγλικα, αντιστοιχες αγγλικα, ιδιος τιμη, κλειδι τιμη, χρονο εκτελεση

Θέμα	Όροι
8	χρηστης δινω, νικου λιστε, πλεονεκτημας κωδικο, αντιστοιχη αγγλικη, συνωνυμος λεξεο
9	αντιστοιχη αγγλικη, αντιστοιχες αγγλικες, τιμη κλειδιο, αναζητηση γινομαι, κλειδι τιμες
10	αντιστοιχη τιμη, τιμες αγγλικες, δινω δυνατοτητας, τιμη αγγλικα, χρονο προσπελαση

Πίνακας 9: LDA 10 θέματα για τελικές απαντήσεις MOOC

Θέμα	Όροι
1	κλειδι τιμη, αντιστοιχη τιμη, χρηση λιστας, χρονο αναζητηση, μεταφραση αγγλικα
2	ανεξαρτη πληθο, σταθερος ανεξαρτη, χρονος προσδιορισμου, προσδιορισμου τιμης, τιμης σταθερος
3	ελληνικης γλωσσα, αγγλικης γλωσσα, φτιαζω κλειδιος, δεδομενο λεξικα, γινομαι αναζητηση
4	τιμες αγγλικες, κλειδιος τιμες, κλειδι τιμες, δυο λιστε, ιδιος τιμη
5	συνθετη δομη, χρησιμοποιησω συνθετη, οδηγω ιδιος, δομη δεδομενο, ελληνικη_λεξη αγγλικη_λεξη
6	τιμη αγγλικη, κλειδι τιμη, αντιστοιχη αγγλικη, μπορουσε γινω, αναζητηση γινομαι
7	τιμη κλειδιο, τιμη value, πιθανες μεταφρασει, χρονο αναζητηση, αντιστοιχες αγγλικες
8	δυο λιστε, κλειδι αγγλικες, αγγλικες τιμες, δυο λεξικα, τιμες κλειδια
9	πληθο δεδομενος, μπορω χρησιμοποιησω, βασικος πλεονεκτημας, μειονεκτημο μπορω, print adict
10	κλειδι μπορω, τιμες λιστε, δηλαδη κλειδι, ελληνικα αγγλικα, αγγλικα τιμες

Πίνακας 10: NMF 10 θέματα για τελικές απαντήσεις MOOC

Θέμα	Όροι
1	σταθερος_ανεξαρτη, ανεξαρτη_πληθο, χρονος_προσδιορισμου, προσδιορισμου_τιμης, τιμης_σταθερος
2	δυο_λιστε, κλειδιος_τιμες, τιμες_αγγλικες, ελληνικης_γλωσσα, αγγλικης_γλωσσα
3	αντιστοιχες_τιμες, simasia_simasia, αγγλικες_τιμες, κλειδι_αγγλικες, κλειδια_δινω

Θέμα	Όροι
4	συνθετη_δομη, χρονο_προσπελαση, μεταφραση_αγγλικα, αντιστοιχη_τιμη, δομη_δεδομενο
5	τιμη_αγγλικη, αντιστοιχη_αγγλικη, κλειδι_τιμη, χρονο_εκτελεση, αγγλικη_γλωσσας
6	δυο_λιστε, κλειδι_μπορω, μπορω_οτιδηποτε, for_range, στοιχειας_λιστας
7	κλειδι_τιμη, χρονο_αναζητηση, δυο_λιστο, δομη_δεδομενο, αγγλικης_ελληνικης
8	κλειδι_τιμη, αναζητηση_γινομαι, χρονο_αναζητηση, τιμη_κλειδιο, γινομαι_βαση
9	χρονος_προσδιορισμου, προσδιορισμου_τιμης, ανεξαρτη_πληθο, langdict_αγγλικη, σταθερος_ανεξαρτη
10	τιμη_κλειδι, ελληνικης_γλωσσα, λιστας_χρηση, συνδυασμο_λιστας, χρηστης_πληκτρολογω

Πίνακας 11: GSDMM 10 θέματα για τελικές απαντήσεις MOOC

Θέμα	Όροι
1	κλειδι_τιμη, δυο_λιστε, αντιστοιχη_αγγλικη, χρονο_αναζητηση, τιμη_αγγλικη
2	σταθερος_ανεξαρτη, χρονος_προσδιορισμου, τιμης_σταθερος, προσδιορισμου_τιμης, ανεξαρτη_πληθο
3	γινομαι_πολυτιμη, εγγενως_απλη, ταχυτερη_αποτελεσματικη, διοτι_εγγενως, χρηση_ταχυτερη
4	κλειδι_υπαρχω, προτεινω_δημιουργιας, ιδιος_νοημο, γινομαι_αντιστοιχιση, χρηστης_εισαγω
5	μπορω_ιδιος, λεξικα_υλοποιω, υλοποιω_πινακας, ιδιος_κλειδι, αντιστοιχες_αγγλικα
6	langdict_αγγλικη, print_adict, δινω_κλειδι, συγκεκριμενος_προβλημας, πλεονεκτημας_εναντι
7	αντιστοιχες_αγγλικες, αγγλικες_μεταφρασει, κλειδι_τιμες, χρηση_κλειδι, τιμες_αντιστοιχες
8	αγγλικης_γλωσσα, μπορω_αλλαζω, κλειδι_μπορω, δηλαδη_κλειδι, δεδομενο_λιστας
9	χρονο_προσπελαση, ελληνικη_λεξη, μορφη_ελληνικη_λεξη, οδηγω_ιδιος, κλειδι_δηλαδη
10	χρησιμοποιησω_λιστε, κλειδια_δινω, χρηστης_δινω, αντιστοιχες_τιμες, κανω_κωδικο

Πίνακας 12: SeaNMF 10 θέματα για τελικές απαντήσεις MOOC

Παρατηρήσεις:

- LDA: Τα θέματα 5 και 9 αφορούν τις ελληνικές και αγγλικές τιμές. Το θέμα 3 φαίνεται και αυτό να έχει καλή συλλογή λέξεων που πιθανότητα αφορούν την υλοποίηση του αλγορίθμου.
- NMF: Τα θέματα 4, 8,10 μιλάνε για τις τιμές και το 7 έφτασε κοντά αλλά είχε τον όρο " χρόνο αναζήτηση". Το θέμα 2 μιλάει είτε για υλοποίηση είτε για πολυπλοκότητα.
- GSDMM: Έχει τα θέμα 1, 6 και 8 που είναι η υλοποίηση. Το θέμα 10 αφορά τις λίστες και τα κλειδιά
- SeaNMF: Τα θέμα 2,4 και ίσως 5 ανήκουν στην υλοποίηση. Το θέμα 7 αφορά τις τιμές/κλειδιά

4.5.3 Πληθυσμός SPOC με 5 θέματα

Θέμα	Όροι
1	συνδυασμο λιστας, χωρο μνημη, αντιστοιχη αγγλικη, χρηση δυο, αντιστοιχες αγγλικες
2	αντιστοιχη αγγλικη, χρονο εκτελεση, εκτελεση προγραμμα, χρονο εντοπισμο, συνθετη δομη
3	αντιστοιχες αγγλικες, αντιστοιχες αγγλικα, χρονο εκτελεση, τιμες αντιστοιχες, χρονο εντοπισμο
4	τιμες λιστε, λιγοτερη μνημη, μεταφρασει αγγλικα, τιμη κλειδιο, μνημη υπολογιστη
5	χωρο μνημη, αντιστοιχη τιμη, λιγοτερος χρονο, κυριο πλεονεκτημας, χρονο προσπελαση

Πίνακας 13: LDA 5 θέματα για τελικές απαντήσεις SPOC

Θέμα	Όροι
1	αντιστοιχες αγγλικες, χρονο εκτελεση, αντιστοιχες αγγλικα, εκτελεση προγραμμα, αντιστοιχη τιμη
2	αντιστοιχη αγγλικη, ελληνικης γλωσσα, καλυτερος τροπο, ελληνικε αγγλικες, αντιστοιχες αγγλικης
3	αντιστοιχη αγγλικη, ελληνικης γλωσσα, καλυτερος τροπο, ελληνικε αγγλικες, αντιστοιχες αγγλικης
4	συνδυασμο λιστας, δυο λιστε, ευρεση αγγλικης, λιστε προτεινω, υπαρχω κλειδι

Θέμα	Όροι
5	μεταφρασει αγγλικα, τιμες λιστε, δομη δεδομενο, κλειδι αντιστοιχω, προσπελαση δεδομενος

Πίνακας 14: NMF 5 θέματα για τελικές απαντήσεις SPOC

Θέμα	Όροι
1	αντιστοιχη_αγγλικη, δυο_λιστε, χρονο_εκτελεση, κλειδιος_τιμες, αντιστοιχες_αγγλικες
2	αντιστοιχη_αγγλικη, ελληνικης_γλωσσα, λιγοτερη_μνημη, συνδυασμο_λιστας, δομη_δεδομενο
3	χωρο_μνημη, λιγοτερος_χρονο, μεταφρασει_αγγλικα, σχεση_λιστε, χρονο_αυξανομαι
4	αντιστοιχες_αγγλικα, αντιστοιχες_αγγλικες, κλειδι_αντιστοιχω, τιμη_κλειδιο, χρηση_δυο
5	χρονο_εντοπισμο, χρονο_προσπελαση, αυξανομαι_χρονο, αντιστοιχες_αγγλικες, αυξανομαι_πληθο

Πίνακας 15: GSDMM 5 θέματα για τελικές απαντήσεις SPOC

Θέμα	Όροι
1	αντιστοιχη_αγγλικη, συνδυασμο_λιστας, χρονο_εκτελεση, αντιστοιχες_αγγλικα, μεταφρασει_αγγλικα
2	χρονο_χρειαζομαι, συνθετης_δομης, τιμη_κλειδιο, κυριο_πλεονεκτημας, δομη_δεδομενο
3	χωρο_μνημη, χρονο_εκτελεση, λιγοτερος_χρονο, ελληνικης_γλωσσα, αντιστοιχη_αγγλικη
4	αντιστοιχες_αγγλικες, αντιστοιχη_τιμη, τιμες_αντιστοιχες, χρονο_εντοπισμο, χρηση_λιστας
5	τιμες_λιστε, χρονο_προσπελαση, χρηστης_πληκτρολογω, κλειδι_τιμες, χρηση_λεξικων

Πίνακας 16: SeaNMF 5 θέματα για τελικές απαντήσεις SPOC

Παρατηρήσεις:

- LDA: Πέρα από το θέμα 5 που μιλάει για πολυπλοκότητα, όλα τα άλλα θέματα έχουν παρόμοιες λέξεις μεταξύ τους.

- NMF: Τα θέματα 2 και 3 μιλάνε για αγγλικές/ελληνικές τιμές. Το θέμα 4 φαίνεται να έχει νέες ενδιαφέρων λέξεις αλλά πάλι είναι μία μίξη διάφορων θεματικών.
- GSDMM: Το θέμα 4 αφορά τις αγγλικές τιμές. Μπορούμε να δούμε ότι όλα τα θέματα αφορούν μόνο τις γενικές θεματικές τιμές/κλειδιά και πολυπλοκότητα.
- SeaNMF: Το θέμα 2 αφορά την υλοποίηση.

4.5.4 Πληθυσμός MOOC με 5 θέματα

Θέμα	Όροι
1	σταθερος ανεξαρτη, ανεξαρτη πληθο, χρονος προσδιορισμου, προσδιορισμου τιμης, τιμης σταθερος
2	κλειδι τιμη, αντιστοιχες αγγλικα, αντιστοιχη αγγλικη, συνθετη δομη, αντιστοιχες αγγλικες
3	κλειδι τιμη, τιμη κλειδι, αντιστοιχη αγγλικα, τιμη αγγλικη, γινομαι αναζητηση
4	δυο λιστε, τιμες αγγλικες, αγγλικη γλωσσας, κλειδιος τιμες, δυο λεξικα
5	αγγλικης γλωσσα, ελληνικης γλωσσα, πιθανες μεταφρασει, γλωσσα τιμες, περισσοτερος χρονο

Πίνακας 17: LDA 5 θέματα για τελικές απαντήσεις MOOC

Θέμα	Όροι
1	κλειδι τιμη, τιμη αγγλικη, αντιστοιχη αγγλικη, χρονο αναζητηση, χρονο εκτελεση
2	ανεξαρτη πληθο, σταθερος ανεξαρτη, χρονος προσδιορισμου, προσδιορισμου τιμης, τιμης σταθερος
3	ελληνικης γλωσσα, αγγλικης γλωσσα, μπορουσε χρηση, γινομαι αναζητηση, γλωσσα τιμες
4	δυο λιστε, τιμες αγγλικες, κλειδιος τιμες, κλειδι τιμες, αγγλικη γλωσσας
5	συνθετη δομη, δυο λεξικα, κλειδι αγγλικες, μεταφραση αγγλικα, βασικος πλεονεκτημας

Πίνακας 18: NMF 5 θέματα για τελικές απαντήσεις MOOC

Θέμα	Όροι
1	τιμες_αγγλικες, κλειδιος_τιμες, δυο_λιστε, κλειδι_μπορω, αντιστοιχες_αγγλικα

Θέμα	Όροι
2	δυο_λιστε, αγγλικης_γλωσσα, προσπελαση_δεδομενος, συνθετη_δομη, ελληνικης_γλωσσα
3	ανεξαρτη_πληθο, χρονος_προσδιορισμου, προσδιορισμου_τιμης, σταθερος_ανεξαρτη, τιμης_σταθερος
4	κλειδι_τιμη, τιμη_αγγλικη, αντιστοιχη_αγγλικη, χρονο_αναζητηση, χρονο_εκτελεση
5	κλειδι_τιμη, αντιστοιχη_αγγλικη, γινομαι_αναζητηση, ελληνικης_γλωσσα, τιμη_κλειδι

Πίνακας 19: GSDMM 5 θέματα για τελικές απαντήσεις MOOC

Θέμα	Όροι
1	ανεξαρτη_πληθο, σταθερος_ανεξαρτη, κλειδι_τιμη, χρονος_προσδιορισμου, προσδιορισμου_τιμης
2	αντιστοιχες_τιμες, δημιουργιας_κλειδια, τιμες_αντιστοιχες, χρησιμοποιησω_λιστε, αντιστοιχες_αγγλικες
3	γινομαι_πολυτιμη, χρηση_ταχυτερη, διοτι_εγγενως, ταχυτερη_αποτελεσματικη, απλη_χρηση
4	μπορω_ιδιος, ιδιος_αγγλικη, πινακας_κατακερματισμου, λεξικα_υλοποιω, υλοποιω_πινακας
5	κλειδι_ελληνικος, ελληνικος_τιμη, χρηστης_εισαγω, δηλαδη_κλειδι, προτεινω_δημιουργιας

Πίνακας 20: SeaNMF 5 θέματα για τελικές απαντήσεις MOOC

Παρατηρήσεις:

- LDA: Το θέμα 1 μιλάει για τις σταθερές τιμές. Τα θέμα 2,3 και 4 αφορούν περισσότερο τις τιμές/κλειδιά και ίσως λίγο την υλοποίηση.
- NMF: Στο θέμα 2 έχουμε ξανά για σταθερές τιμές. Τα θέματα 3 και 4 είναι για τιμές/κλειδιά.
- GSDMM: Το θέμα 3 αφορά τις σταθερές τιμές και το θέμα 2 υλοποίηση.
- SeaNMF: Τα θέμα 1 και 4 αφορούν την υλοποίηση, το θέμα 2 για τιμές/κλειδιά. Τέλος το θέμα 3 για πολυπλοκότητα.

4.6 ΠΡΟΒΛΗΜΑΤΑ ΑΞΙΟΛΟΓΗΣΗΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Τα περισσότερα μοντέλα θεμάτων λειτουργούν με μάθηση χωρίς επίβλεψη και όπως για κάθε αλγόριθμο χρειάζεται να ξέρουμε αν παράγουν σωστά αποτελέσματα.

Ο πιο εύκολος και γρήγορος τρόπος για να λάβουμε ένα μετρό απόδοσης είναι με μαθηματικά. Μια τέτοια τιμή απόδοσης είναι το perplexity που μας δείχνει μετά την εκπαίδευση πάνω στο corpus ποσό "μπερδεμένο" είναι. Δηλαδή μετράει πόσο καλά οι

όροι του corpus αναπαριστώντες στις κατανομές από τα θέματα. Όσο πιο μικρή είναι η τιμή, τόσοι καλυτέρα το μοντέλο βρίσκει μοτίβα φυσικής γλώσσας. Οι τεχνικές στατιστικής μέτρησης διαφέρουν από μοντέλο σε μοντέλο και δεν οι τιμές τους δεν μας προσφέρουν πολλά. Στη καλύτερη περίπτωση μπορούν να μας βοηθήσουν να επιλέξουμε καλύτερες παραμέτρους για την εκπαίδευση του μοντέλου, πέρα από αυτό δεν αντικατοπτρίζουν το πραγματικό κόσμο.

Για αυτό το λόγο χρησιμοποιούμε αυτά τα μέτρα απόδοσης ως μία απλή ένδειξη ότι το μοντέλο μας με τις συγκεκριμένες μεταβλητές που του έχουμε σώσει δουλεύει σωστά. Από εκεί και πέρα οι περισσότερες τεχνικές που έχουν προταθεί είναι πιο χρονοβόρες και συνήθως απαιτούν αρκετούς ανθρώπους για να βγει ένα έμπιστο αποτέλεσμα.

Οι παρακάτω τεχνικές γίνονται από ανθρώπους και μπορεί να γίνουν υποκειμενικές. Δεν προτείνεται να τις εκτελέσει μόνο ο αναλυτής διότι ενδέχεται να είναι προκατειλημμένος και θέλει το μοντέλο του να δουλέψει. Παράγει μια απεικόνιση των τοπ λέξεων για κάθε θέμα και φτιάχνει συνδέσεις στο μυαλό του για τις σχέσεις των λέξεων που δεν υπάρχουν. Επίσης δεν μπορούμε να μετρήσουμε αν ένα θέμα είναι καλό ή κακό διότι ο καθένας έχει τον όρο "θέμα" διαφορετικά στο μυαλό. Επομένως, για να μειώσουμε προκατάληψη στα αποτελέσματα και να βρούμε το μέσο όρο απόδοσης ζητάμε μια ομάδα ανθρώπων να δοκιμάσει τις παρακάτω τεχνικές.

Στη πρώτη τεχνική που ονομάζεται Word Intrusion δείχνουμε σε ένα άτομο μια λίστα από λέξεις και του ζητάμε να βρει τη λέξη που δεν ταιριάζει με το σύνολο (Chang, 2009). Παίρνουμε τις λέξεις με την υψηλότερη πιθανότητα εμφάνισης από ένα θέμα X και άλλη μια λέξη με χαμηλή πιθανότητα εμφάνισης στο θέμα X αλλά με υψηλή σε ένα άλλο θέμα Y.

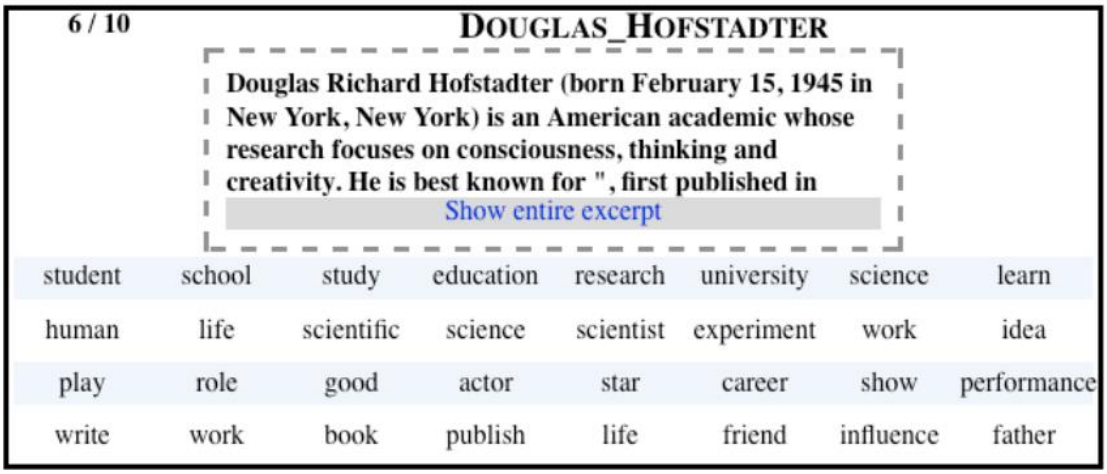
Εικόνα 24: Παράδειγμα Word Intrusion

Στο παράδειγμα μας οι όροι "ήλιος", "καλοκαίρι", "θάλασσα", "παραλία" εμφανίζονται συχνά στο θέμα X ενώ ο όρος "απόδοση" εμφανίζεται ελάχιστα στο θέμα X αλλά εμφανίζεται συχνά στο θέμα Y. Αν ο χρήστης βρει τη σωστή λέξη τότε το μοντέλο φαίνεται να βγάζει καλά αποτέλεσμα. Αν κάνουμε αυτή τη διαδικασία αρκετές φορές μπορούμε να βγάλουμε καλυτέρα συμπεράσματα. Όσο πιο κοντά στο 1 τόσο πιο καλά θέματα παράγει το μοντέλο μας.

Ακρίβεια μοντέλου = (σωστές απαντήσεις/όλες οι απαντήσεις)

Η δεύτερη τεχνική ονομάζεται Topic Intrusion, σε αυτή δείχνουμε σε ένα άτομο ένα από τα έγγραφα μας και μαζί με 4 θέματα (από το κάθε θέμα δείχνουμε τις λέξεις με μεγαλύτερη πιθανότητα εμφάνισης). Τα 3 από τα 4 θέματα έχουν μεγάλη πιθανότητα

να ανατεθούν στο έγγραφο ενώ το 4ο θέμα έχει μικρή πιθανότητα να ανατεθεί στο έγγραφο. (Chang, 2009)



Εικόνα 25: Παράδειγμα Topic Intrusion από Chang και Boyd-Graber (Chang, 2009)

Το άτομο θα πρέπει να βρει το θέμα με τη μικρότερη πιθανότητα εμφάνισης.

ΚΕΦΑΛΑΙΟ 5: ΣΥΜΠΕΡΑΣΜΑΤΑ

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η εργασία αυτή είχε ως στόχους να υλοποιήσουμε μια σειρά διεργασιών (processing pipeline) ώστε να προ-επεξεργαστούμε τα δεδομένα και έπειτα να δοκιμάσουμε τέσσερις διαφορετικούς αλγόριθμους μοντελοποίησης θεμάτων.

Τα βασικά ερωτήματα που δημιουργήθηκαν και επιχειρήθηκε να απαντηθούν στην παρούσα εργασία είναι:

1. Υπήρχαν δυσκολίες στην στατιστική ανάλυση του κειμένου;

   Δεν υπήρχαν σοβαρές δυσκολίες αλλά χρειάστηκε να καθαριστούν κάποια κομμάτια από τα δεδομένα χειροκίνητα βάση των κανόνων που θέσαμε στη Πολιτική Εκκαθάρισης Συνόλου Δεδομένων

2. Υπήρχαν δυσκολίες στη προ-επεξεργασία των εγγράφων;

   Ναι διότι τα έγγραφα μας είναι στα ελληνικά και οι τεχνικές που εφαρμόστηκαν δεν έχουν δοκιμαστεί διεξοδικά στην ελληνική γλώσσα. Προτειχίσαμε να επιλέξουμε έναν πιο εξειδικευμένο lemmatizer που ήταν εκπαιδευμένος σε ελληνικές λέξεις ώστε να έχουμε καλύτερα αποτελέσματα. Επίσης σε αντίθεση με άλλες γλώσσες τα ελληνικά έχουν τόνους και υπάρχουν λέξεις που αν αλλάξουμε τον τόνο αλλάζει τελείως το νόημα της λέξης.

3. Ποιους αλγόριθμους μοντελοποίησης θεμάτων επιλέξατε και γιατί?

   Επιλέξαμε τον αλγόριθμο Latent Dirichlet Allocation (LDA) διότι είναι μια δημοφιλής τεχνικής και τις περισσότερες φορές παράγει ικανοποιητικά αποτελέσματα. Στη συνέχεια επιλέξαμε τον αλγόριθμο Non-negative Matrix Factorization(NMF ή NNMF) που δεν είναι πικάντικός όπως ο LDA. Τέλος προσθέσαμε και τους δύο αλγόριθμους Gibbs sampling for Dirichlet Mixture Model (GSDMM) και Semantics-assisted Non-negative Matrix Factorization (SeaNMF) που είναι εξειδικευμένες εκδοχές του LDA και NMF αντίστοιχα και υπόσχονται καλά αποτελέσματα για μικρά κείμενα όπως οι τελικές απαντήσεις των μαθητών.

4. Τι συμπεραίνουμε από τα αποτελέσματα των αλγορίθμων?

   Αρχικά, παρατηρήσαμε πως ο σχεδόν διπλάσιος αριθμός εγγράφων των δεδομένων (dataset) με απαντήσεις μαθητών του πληθυσμού MOOC βοήθησε στα αποτελέσματα διότι βρήκαμε πιο ισχυρά θέματα που όλοι οι όροι ταίριαζαν στο σύνολο. Κάποιοι όροι όπως το “αντιστοιχες_τιμες” θα μπορούσαμε να το κατατάξουμε σε διάφορα γενικότερα θέματα όπως υλοποίηση και τιμές. Τελικά παρατηρήθηκε πως οι αλγόριθμοι GSDMM και SeaNMF βγάζουν περισσότερα ολοκληρωμένα θέματα σε σχέση με το NMF και ιδιαίτερα με το LDA.

   Συμπεραίνουμε ότι οι αλγόριθμοι GSDMM και SeaNMF ενώ παράγουν καλύτερα αποτελέσματα από LDA και NMF, μπορούν να θεωρηθούν μόνο ως μία υπόδειξη για τη γενική εικόνα των δεδομένων. Πιστεύουμε ότι ο μικρός αριθμός εγγράφων που χρησιμοποιήθηκε δεν ήταν αρκετός ώστε να παράγει αξιοσημείωτα αποτελέσματα.

Τα ανοιχτά ζητήματα που θα μπορούσαν να διερευνηθούν από μια μελλοντική επέκταση της παρούσας εργασίας είναι πως συμπεριφέρονται οι αλγόριθμοι που συγκεντρώσαμε με ένα μεγαλύτερο σύνολο δεδομένων και πως οι αλλαγές στις μεταβλητές των αλγόριθμος επηρεάζουν το αποτέλεσμα.

ΠΑΡΑΡΤΗΜΑ Ι: ΑΝΑΦΟΡΕΣ

ΒΙΒΛΙΟΓΡΑΦΙΑ

Blei, David M., et al. “Latent Dirichlet Allocation.” The Journal of Machine Learning Research, vol. 3, no. null, Mar. 2003, pp. 993–1022.

Brinton, C. G., et al. “Learning about Social Learning in MOOCs: From Statistical Analysis to Generative Model.” IEEE Transactions on Learning Technologies, vol. 7, no. 4, Oct. 2014, pp. 346–59. IEEE Xplore, doi:10.1109/TLT.2014.2337900.

Chang, Jonathan, et al. “Reading Tea Leaves: How Humans Interpret Topic Models.” Proceedings of the 22nd International Conference on Neural Information Processing Systems, Curran Associates Inc., 2009, pp. 288–96.

Chen, Ye, et al. “Topic Modeling for Evaluating Students’ Reflective Writing: A Case Study of Pre-Service Teachers’ Journals.” Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, Association for Computing Machinery, 2016, pp. 1–5. ACM Digital Library, doi:10.1145/2883851.2883951.

Gottipati, Swapna, et al. “Text Analytics Approach to Extract Course Improvement Suggestions from Students’ Feedback.” Research and Practice in Technology Enhanced Learning, vol. 13, no. 1, June 2018, p. 6. Springer Link, doi:10.1186/s41039-018-0073-0.

Mikolov, Tomas, et al. “Efficient Estimation of Word Representations in Vector Space.” ArXiv:1301.3781 [Cs], Sept. 2013. arXiv.org, http://arxiv.org/abs/1301.3781.

Resnik, Philip, et al. “Using Topic Modeling to Improve Prediction of Neuroticism and Depression in College Students.” Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2013, pp. 1348–53. ACLWeb, https://www.aclweb.org/anthology/D13-1133.

Sarkar, Dipanjan. Text Analytics with Python: A Practitioner’s Guide to Natural Language Processing. Apress, 2019. DOI.org (Crossref), doi:10.1007/978-1-4842-4354-1.

Shi, Tian, et al. “Short-Text Topic Modeling via Non-Negative Matrix Factorization Enriched with Local Word-Context Correlations.” Proceedings of the 2018 World Wide Web Conference, International World Wide Web Conferences Steering Committee, 2018, pp. 1105–14. ACM Digital Library, doi:10.1145/3178876.3186009.

Shukor, Nurbiha A., and Zaleha Abdullah. “Using Learning Analytics to Improve MOOC Instructional Design.” International Journal of Emerging Technologies in Learning (IJET), vol. 14, no. 24, Dec. 2019, pp. 6–17.

Vytasek, Jovita M., et al. “Topic Models to Support Instructors in MOOC Forums.” Proceedings of the Seventh International Learning Analytics & Knowledge Conference, Association for Computing Machinery, 2017, pp. 610–11. ACM Digital Library, doi:10.1145/3027385.3029486.

Wang, Ling, et al. “Semantic Analysis of Learners’ Emotional Tendencies on Online MOOC Education.” Sustainability, vol. 10, no. 6, June 2018, p. 1921. DOI.org (Crossref), doi:10.3390/su10061921.

Yin, Jianhua, and Jianyong Wang. “A Dirichlet Multinomial Mixture Model-Based Approach for Short Text Clustering.” Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, 2014, pp. 233–42. ACM Digital Library, doi:10.1145/2623330.2623715.

Zhao, Wayne Xin, et al. “Comparing Twitter and Traditional Media Using Topic Models.” Advances in Information Retrieval, edited by Paul Clough et al., Springer, 2011, pp. 338–49. Springer Link, doi:10.1007/978-3-642-20161-5_34.

Δημητριάδης, Νικόλαος. Applying Topic Modelling Algorithms on Twitter Messages in Greek Language. 2020.

WEB SITES

[1] "By The Numbers: MOOCs in 2020 — Class Central." The Report by Class Central, 30 Nov. 2020, https://www.classcentral.com/report/mooc-stats-2020/ (τελευταία επίσκεψη 11 Μαρτίου 2021)

[2] Chaudhary, Mukesh. "TF-IDF Vectorizer Scikit-Learn." Medium, 28 Jan. 2021, https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a (τελευταία επίσκεψη 11 Μαρτίου 2021)

[3] Dietz, Laura. Topic Model Evaluation: How Much Does It Help? 2016, http://topicmodels.info/ckling/tmt/part4.pdf (τελευταία επίσκεψη 11 Μαρτίου 2021)

[4] Eellak/Gsoc2018-Spacy. 2018. GFOSS - Open Technologies Alliance (Οργανισμός Ανοιχτών Τεχνολογιών - ΕΕΛΛΑΚ), 2021. GitHub, https://github.com/eellak/gsoc2018-spacy (τελευταία επίσκεψη 11 Μαρτίου 2021)

[5] "Introduction to Topic Modeling." MonkeyLearn Blog, 26 Sept. 2019, https://monkeylearn.com/blog/introduction-to-topic-modeling/ (τελευταία επίσκεψη 11 Μαρτίου 2021)

[6] LaptrinhX, "Introduction to Stemming vs Lemmatization (NLP)." LaptrinhX, 4 Sept. 2020, https://laptrinhx.com/introduction-to-stemming-vs-lemmatization-nlp-1503911125/. (τελευταία επίσκεψη 11 Μαρτίου 2021)

[7] Matti Lyra - Evaluating Topic Models. www.youtube.com, https://www.youtube.com/watch?v=UkmIljRIG_M (τελευταία επίσκεψη 11 Μαρτίου 2021)

[8] "Necessary to Apply TF-IDF to New Documents in Gensim LDA Model?" Stack Overflow, https://stackoverflow.com/questions/44781047/necessary-to-apply-tf-idf-to-new-documents-in-gensim-lda-model (τελευταία επίσκεψη 11 Μαρτίου 2021)

[9] Topic Modeling Guide (GSDM,LDA,LSI). https://kaggle.com/ptfrwrd/topic-modeling-guide-gsdm-lda-lsi (τελευταία επίσκεψη 11 Μαρτίου 2021)

[10] Topic Modelling In Python Using Latent Semantic Analysis." Analytics Vidhya, 1 Oct. 2018, https://www.analyticsvidhya.com/blog/2018/10/stepwise-guide-topic-modeling-latent-semantic-analysis/ (τελευταία επίσκεψη 11 Μαρτίου 2021)

[11] Prateek, Joshi. "Text Mining 101: A Stepwise Introduction to Topic Modeling using Latent Semantic Analysis (using Python)" Analytics Vidhya, 1 Oct. 2018, https://www.analyticsvidhya.com/blog/2018/10/stepwise-guide-topic-modeling-latent-semantic-analysis/ (τελευταία επίσκεψη 11 Μαρτίου 2021)

[12] Xu, Joyce. "Topic Modeling with LSA, PSLA, LDA & Lda2Vec." Medium, 20 Dec. 2018, https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05 (τελευταία επίσκεψη 11 Μαρτίου 2021)

ΠΑΡΑΡΤΗΜΑ ΙΙ: ΚΩΔΙΚΑΣ

ΚΩΔΙΚΑΣ

GitHub

ΠΑΡΑΡΤΗΜΑ ΙΙΙ: ΑΚΡΩΝΥΜΑ

ΕΛΛΗΝΙΚΟΙ ΟΡΟΙ

Ακρώνυμο	Επεξήγηση
ΜΑΔΜ	Μαζικά Ανοικτά Διαδικτυακά Μαθήματα

ΑΓΓΛΙΚΟΙ ΟΡΟΙ

Ακρώνυμο	Επεξήγηση
MOOC	Massive Open Online Course
SPOC	Small Private Online Course
LDA	Latent Dirichlet Allocation
NMF/NNMF	Non-negative Matrix Factorization
GSDMM	Gibbs sampling for Dirichlet Mixture Model
SeaNMF	Semantics-assisted Non-negative Matrix Factorization
TF-IDF	Term Frequency – Inverse Document Frequency

ΠΑΡΑΡΤΗΜΑ IV: ΓΛΩΣΣΑΡΙΟ

ΓΛΩΣΣΑΡΙΟ

Όρος	Επεξήγηση
Dataset	Δεδομένα ή Σύνολο δεδομένων
Session	Συνεδρία
Agent	Πράκτορας
Topic Modeling	Μοντελοποίηση Θεμάτων
Lemmatization	Διεργασία μετατροπής μιας λέξης στη βασική της μορφή
Stopwords	Κοινές λέξεις που δεν επηρεάζουν το νόημα του κειμένου και μπορούν να αφαιρεθούν.
Bag-of-words	Αναπαράσταση των εγγράφων ως ένας πίνακας λέξεων που η γραμμές είναι τα έγγραφα μας και οι στήλες όλα τα μοναδικά tokens
Token	Λέξεις, φράσεις, αριθμοί και άλλα σύμβολα
CountVectorizer	Μοντέλο Bag-of-words που αθροίζει τις εμφανίσεις ενός όρου
TfidfVectorizer	Μοντέλο Bag-of-words που υπολογίζει μια σταθμισμένη τιμή για τις εμφανίσεις ενός όρου

ΠΑΡΑΡΤΗΜΑ V: ΕΥΡΕΤΗΡΙΟ

ΕΥΡΕΤΗΡΙΟ