2017

Ανάκτηση Πληροφορίας: Opinion mining

ΑΝΑΓΝΩΣΤΟΥ ΑΝΤΩΝΙΟΣ 2268
ΛΑΣΚΑΡΙΔΗΣ ΣΤΕΦΑΝΟΣ 2315
{ANAGNOAD,LASKSTEF}@CSD.AUTH.GR

Περιεχόμενα

Εισαγωγή

Η συγκεκριμένη εργασία εκπονήθηκε στο πλαίσιο του μαθήματος «Ανάκτηση Πληροφορίας», του 7ου εξαμήνου του Τμήματος Πληροφορικής Α.Π.Θ.

Στόχος της εργασίας είναι η μελέτη και υλοποίηση τεχνικών «Opinion mining», πάνω σε σύνολα κριτικών ταινιών από χρήστες. Κάθε κριτική μπορεί να χαρακτηριστεί είτε ως «θετική», είτε «αρνητική».

Το τεχνικό μέρος της εργασίας αναπτύχθηκε σε Apache Spark 2.0.1 και Java 1.8. Οι αλγόριθμοι που εξετάστηκαν διακρίνονται σε supervised και unsupervised και βρίσκονται εγγενώς υλοποιημένοι στο Spark. Για την προεπεξεργασία των δεδομένων, χρησιμοποιήθηκε, επίσης, ένα Python 3 script.

Για την εκτέλεση τόσο της προεπεξεργασίας των δεδομένων όσο και των αλγορίθμων που περιγράφονται παρακάτω, συμβουλευτείτε το αρχείο README.md που περιέχει τις εντολές που χρειάζεται να τρέξετε.

Μετασχηματισμός δεδομένων

Αρχικά, για τη φόρτωση των δεδομένων μας στο spark, μετασχηματίσαμε τα δεδομένα μας σε ένα json αρχείο στο οποίο ενσωματώνεται το κείμενο, το id καθώς και τα stars και label προκειμένου να έχουμε schema στο dataset του Spark.

Για την αναπαράσταση των δεδομένων χρησιμοποιήθηκε το διανυσματικό μοντέλο, με χρήση των μετρικών $tf * idf$ ως βάρη. Για τον διαχωρισμό των κειμένων σε όρους, χρησιμοποιούμε μία κανονική έκφραση προκειμένου να διασπαστεί το κείμενο τόσο με βάση τα κένα όσο και με βάση τα σημεία στίξης.

Για την αύξηση της αξιοπιστίας των μοντέλων, αφαιρούμε τα εξής stopwords από τους όρους:

"i","me","my","myself","we","our","ours","ourselves","you","your","yours","yourself","yourselves","he","him","his","himself","she","her","hers","herself","it","its","itself","they","them","their","theirs","themselves","what","which","who","whom","this","that","these","those","am","is","are","was","were","be","been","being","have","has","had","having","do","does","did","doing","a","an","the","and","but","if","or","because","as","until","while","of","at","by","for","with","about","against","between","into","through","during","before","after","above","below","to","from","up","down","in","out","on","off","over","under","again","further","then","once","here","there","when","where","why","how","all","any","both","each","few","more","most","other","some","such","only","own","same","so","than","too","very","s","t","can","will","just","don","should","now","d","ll","m","o","re","ve","y"

Επιπλέον, χρησιμοποιούμε 2-grams, προκειμένου να μειωθούν οι σημασιολογικές απώλειες.

Για την αύξηση της ακρίβειας των αλγορίθμων, χρησιμοποιήθηκε επιπλέον το λεξικό θετικών/αρνητικών όρων SemanticNet4, προκειμένου να αυξηθούν τα βάρη των λέξεων που προσδίδουν σημασιολογική πληροφορία στις κριτικές. Το λεξικό αυτό προσδίδει ένα βάρος στις λέξεις, στο διάστημα [-1,1] ανάλογα με την σημαντικότητα τους. Δεδομένου ότι δεν μπορούμε να έχουμε πρόσβαση στον αντεστραμμένο κατάλογο για το διανυσματικό μοντέλο, προβήκαμε στην τεχνητή αλλαγή των βαρών βάσει συχνότητας, αυξάνοντάς την, προσθέτοντας στο τέλος κάθε κειμένου κριτικής λέξεις με υψηλό σημασιολογικό βάρος που ήδη εμφανίζονται μέσα στο κείμενο. Συγκεκριμένα, ακολουθήσαμε την εξής τακτική:

Βάρος λέξης	**Αριθμός εισαγόμενων λέξεων**
0 – 0.65	0
0.65 – 0.7	2
0.7 – 0.8	4
0.8 – 0.85	8
0.85 – 0.9	16
0.9 – 1	32

Με αυτό τον τρόπο δίνουμε όλο και μεγαλύτερη σημασία σε σημασιολογικά σημαντικούς όρους, αυξάντοντας την ποσότητα tf, ενώ η ποσότητα idf παραμένει σταθερή.

Για την μείωση των διαστάσεων στους αλγορίθμους μηχανικής μάθησης χωρίς επίβλεψη, οι οποίοι βασίζονται στην απόσταση, χρησιμοποιήθηκε ο αλγόριθμος Word2Vec, ο οποίος προβαίνει σε αλλαγή βάσης του διανυσματικού χώρου από λέξεις σε έννοιες. Ωστόσο, πειραματικά, η εισαγωγή της αναπαράστασης, σε διαφορετικά πλήθη εννοιών, δεν βοήθησε στην αύξηση της ακρίβειας.

Αλγόριθμοι μηχανικής μάθησης με επίβλεψη

Λογιστική Παλινδρόμηση

Εφαρμόζοντας την μέθοδο της λογιστικής παλινδρόμησης (Logistic Regression) και χρησιμοποιόντας 3-fold cross validation, παρατηρήσαμε τα ακόλουθα αποτελέσματα:

Παρατηρήσεις	Αριθμός διαστάσεων	Ακρίβεια
Χρήση σημασιολογικού λεξικού, αναπαράσταση $tf * idf$	Όλες	0.93
	1000	0.76
Αναπαράσταση $tf * idf$ χωρίς την χρήση σημασιολογικού λεξικού, αλλά με την εισαγωγή 2-grams	50	0.93

Μηχανή Υποστήριξης Διανυσμάτων

Εκπαιδεύοντας μία μηχανή υποστήριξης διανυσμάτων (Support Vector Machine) και χρησιμοποιώντας hold-out test validation (εκπαίδευση στο 90% του συνόλου και επαλήθευση στο 10%), παρατηρήσαμε τα ακόλουθα αποτελέσματα:

Παρατηρήσεις	Αριθμός διαστάσεων	Ακρίβεια
Χρήση σημασιολογικού λεξικού, αναπαράσταση $tf * idf$	Όλες	0.72
	1000	0.54
Αναπαράσταση $tf * idf$ χωρίς την χρήση σημασιολογικού λεξικού, αλλά με την εισαγωγή 2-grams	50	0.72

Αλγόριθμοι μηχανικής μάθησης χωρίς επίβλεψη

Αλγόριθμος Κ-μέσων

Εφαρμόζοντας τον αλγόριθμο Κ-μέσων και χρησιμοποιώντας hold-out test validation (εκπαίδευση στο 90% του συνόλου και επαλήθευση στο 10%), παρατήρησαμε τα ακόλουθα αποτελέσματα.

Παρατηρήσεις	Αριθμός διαστάσεων	Ακρίβεια

Χρήση σημασιολογικού λεξικού, αναπαράσταση $tf * idf$	Όλες	0.51
	1000	0.51
Αναπαράσταση Word2Vec χωρίς χρήση σημασιολογικού λεξικού, αλλά με την εισαγωγή 2-grams	50	0.53

Λανθάνουσα κατανομή Dirichlet (LDA)

Εφαρμόζοντας τον αλγόριθμο της Λανθάνουσας κατανομής Dirichlet (LDA) και χρησιμοποιόντας hold-out test validation (εκπαίδευση στο 90% του συνόλου και επαλήθευση στο 10%), παρατήρησαμε τα ακόλουθα αποτελέσματα.

Παρατηρήσεις	**Αριθμός διαστάσεων**	**Ακρίβεια**
Χρήση σημασιολογικού λεξικού, αναπαράσταση $tf * idf$	Όλες	0.52
	1000	0.52
Αναπαράσταση Word2Vec χωρίς χρήση σημασιολογικού λεξικού, αλλά με την εισαγωγή 2-grams	50	N/A - Too long to run

Gaussian Mixture Model (GMM)

Ο αλγόριθμος Gaussian Mixture Model (GMM) τόσο σε 262.144 διαστάσεις, όσο και σε μειωμένο αριθμό διαστάσεων (1000) με την αναπαράσταση $tf * idf$ δεν έδωσε κάποια αποτελέσματα (too long to run). Με χρήση της αναπαράστασης Word2Vec (50 διαστάσεις), ο αλγόριθμος παρουσίασε ακρίβεια 0.50.

Ερμηνεία – Συμπεράσματα

Βάσει των αποτελεσμάτων ακρίβειας των παραπάνω τεχνικών, προτείνουμε τη χρήση λογιστικής παλινδρόμησης προκειμένου να προβλέψουμε τα labels του test set.

Συγκεκριμένα, οι αλγόριθμοι επιβλεπόμενης μάθησης κρίνουμε πως ανταποκρίνονται καλύτερα στις ανάγκες των ζητούμενων του προβλήματος, μιας και λαμβάνουν υπόψη τους τα ήδη υπάρχονται labeled datasets.

Εν αντιθέσει, οι αλγόριθμοι μη επιβλεπόμενης μάθησης, στην προσέγγισή μας, απλώς διακρίνουν το σύνολο σε δύο υπο-ομάδες, ελπίζοντας πως αυτές είναι θετικές/αρνηντικές. Ωστόσο η διάκριση βάσει των χαρακτηριστικών μας μπορεί να είναι τελείως διαφορετική, αγγίζοντας έτσι τα όρια της τυχαιότητας (ακρίβεια=50%). Επιπλέον, οι αλγόριθμοι που βασίζονται σε μέτρα απόστασης φαίνεται να μη λειτουργούν κάτω από συνθήκες μεγάλης διαστασιμότητας, καθώς η έννοια της απόστασης χάνεται. Οποιαδήποτε τυχαία σημεία στο χώρο θα απέχουν πολύ στις 1000+ διαστάσεις.

Τέλος, ακόμη και στους κατά βαση μη επιβλεπόμενους αλγορίθμους, χρησιμοποιούμε το majority vote προκειμένου να μπορέσουμε να χαρακτηρίσουμε την κάθε ομάδα ως «θετική» ή «αρνητική». Διαφορετικά, το μοναδικό συμπέρασμα θα ήταν η διάκριση των συστάδων.

Μελλοντικές επεκτάσεις

Στο άρθρο των Rothfels και Tibshirani περιγράφεται μία τεχνική σημασιολογικής επισήμανσης των δεδομένων η οποία μελετάται μερικώς από εμάς μέσω της ενσωμάτωσης του λεξικού στα βάρη των λέξεων. Λόγω χρονικών περιορισμών δεν προχωρήσαμε σε περεταίρω ανάλυση της συγκεκριμένης προσέγγισης, η οποία θα ήταν η επόμενη κατεύθυνσή μας στο τμήμα της μη επιβλεπόμενης μάθησης.

Βιβλιογραφία

Clustering Algorithms - Gaussian Mixture Model. (n.d.). Ανάκτηση από Apache Spark 2.0.1 Documentation: https://spark.apache.org/docs/2.0.1/ml-clustering.html#gaussian-mixture-model-gmm

Clustering Algorithms - K-Means. (n.d.). Ανάκτηση από Apache Spark 2.0.1 Documentation: https://spark.apache.org/docs/2.0.1/ml-clustering.html#k-means

Clustering Algorithms - LDA. (n.d.). Ανάκτηση από Apache Spark 2.0.1 Documentation: https://spark.apache.org/docs/2.0.1/ml-clustering.html#latent-dirichlet-allocation-lda

Feature Extraction and Transformation - TF-IDF. (n.d.). Ανάκτηση από Apache Spark 2.0.1 Documentation: https://spark.apache.org/docs/2.0.1/mllib-feature-extraction.html#tf-idf

Feature Extraction and Transformation - Word2Vec. (n.d.). Ανάκτηση από Apache Spark 2.0.1 Documentation: https://spark.apache.org/docs/2.0.1/mllib-feature-extraction.html#word2vec

MIT Media Laboratory, *SenticNet.* Ανάκτηση από http://sentic.net/

Linear Classification Methods - Logistic Regression. (n.d.). Ανάκτηση από Apache Spark 2.0.1 Documentation: https://spark.apache.org/docs/2.0.1/mllib-linear-methods.html#logistic-regression

Linear Classification Methods - Support Vector Machines. (n.d.). Ανάκτηση από Apache Spark 2.0.1 Documentation: https://spark.apache.org/docs/2.0.1/mllib-linear-methods.html#linear-support-vector-machines-svms

Rothfels, J. T. (2010). Unsupervised sentiment classification of English movie reviews.