2η Προγραμματιστική Εργασία

Κατηγοριοποίηση Κειμένου

Προεπεξεργασία

Για τη λύση του προβλήματος κατηγοριοποίησης κειμένου και συγκεκριμένα για την προεπεξεργασία του κειμένου χρησιμοποιήθηκαν τεχνικές όπως tokenizing, stemming και stop words filtering.

1. **Tokenize**
   Ουσιαστικά, διασπούμε το κείμενο σε ξεχωριστές μονάδες, συνήθως λέξεις ή φράσεις, γνωστές ως tokens. Το tokenization έχει μεγάλη σημασία, καθώς αποτελεί τα θεμέλια για την μετέπειτα ανάλυση του κειμένου παρέχοντας μια δομημένη αναπαράσταση αυτού. Για να το πετύχουμε αυτό, χρησιμοποιήσαμε στο περιβάλλον του λογισμικού Rapid Miner τον Tokenize Operator παράλληλα με τον Filter Tokens. Στον κώδικα που γράψαμε εμείς, χρησιμοποιήθηκε η συνάρτηση word_tokenize από τη βιβλιοθήκη NLTK, ενώ έγινε και filtering ώστε οποιοιδήποτε χαρακτήρες δεν αποτελούν γράμμα να μη λαμβάνονται υπόψιν.

2. **Stemming**
   Το Stemming είναι η διαδικασία αναγωγής των λέξεων στη ρίζα τους, με στόχο να καταλάβουμε το βασικό νόημα μιας λέξης. Με την εφαρμογή του stemming, οι παραλλαγές των λέξεων απλοποιούνται σε μια κοινή μορφή, μειώνοντας τη διάσταση του dataset. Αυτό είναι επωφελές για την ανάλυση συναισθήματος, καθώς βοηθά στην αναγνώριση λέξεων που περιέχουν συναισθήματα σε διαφορετικές γραμματικές μορφές. Στο Rapid Miner, για να επιτευχθεί αυτό, χρησιμοποιήθηκε το Stem operator και συγκεκριμένα ο Porter Stemmer. Στον δικό μας κώδικα χρησιμοποιήθηκε ο Porter Stemmer από τη βιβλιοθήκη NLTK.

3. **Stopwords filtering**

   Τα stopwords είναι κοινές λέξεις σε μια γλώσσα που δεν έχουν σημαντικό νόημα και συχνά αποκλείονται από την ανάλυση. Με την αφαίρεση λέξεων όπως "the", "and", και "is" το στάδιο της προεπεξεργασίας αποσκοπεί στην εστίαση σε λέξεις που περιέχουν περιεχόμενο και είναι περισσότερο ενδεικτικές του συναισθήματος. Αυτό συμβάλλει στην αποτελεσματικότητα και την ακρίβεια των μετέπειτα μοντέλων κατηγοριοποίησης. Στο Rapid Miner έγινε χρήση του Filter Stopwords (English) Operator και στον κώδικα που γράψαμε η λίστα των stopwords της βιβλιοθήκης NLTK χρησιμοποιήθηκε για το φιλτράρισμα αυτών των κοινών αγγλικών stopwords.

Αυτά τα βήματα προεπεξεργασίας συμβάλλουν συλλογικά στη δημιουργία μιας πιο συνοπτικής, ουσιαστικής και αντιπροσωπευτικής κειμενικής αναπαράστασης των κριτικών ταινιών, βελτιώνοντας τελικά την απόδοση των μοντέλων ανάλυσης συναισθήματος.

Rapid Miner

- ***Binary term occurrences***

accuracy: 75.30% +/- 3.40% (micro average: 75.30%)

	true neg	true pos	class precision
pred. neg	788	282	73.64%
pred. pos	212	718	77.20%
class recall	78.80%	71.80%	

weighted_mean_recall: 75.30% +/- 3.40% (micro average: 75.30%), weights: 1, 1

weighted_mean_precision: 75.62% +/- 3.39% (micro average: 75.42%), weights: 1, 1

- ***Term occurrences***

accuracy: 73.15% +/- 4.11% (micro average: 73.15%)

	true neg	true pos	class precision
pred. neg	781	318	71.06%
pred. pos	219	682	75.69%
class recall	78.10%	68.20%	

weighted_mean_recall: 73.15% +/- 4.11% (micro average: 73.15%), weights: 1, 1
weighted_mean_precision: 73.46% +/- 4.11% (micro average: 73.38%), weights: 1, 1

Σχολιασμός: Παρατηρούμε ότι, μετά και την προεπεξεργασία που έχουμε κάνει, το accuracy μεταξύ των μοντέλων binary term occurrences και term occurrences διαφέρει κατά περίπου 2% (75.3% έναντι 73.15%). Το weighted mean recall στο πρώτο μοντέλο υπολογίζεται στα 75.3% σε αντίθεση με το δεύτερο στα 73.46% (περίπου 2% χαμηλότερα και πάλι). Τέλος το weighted mean recall στο πρώτο 75.62% και το δεύτερο στα 73.15% (περίπου 2.5% η διαφορά).

Χρήση κώδικα σε Python

```
---Classification Results---
Multinomial Naive Bayes Accuracy: 0.80
Bernoulli Naive Bayes Accuracy: 0.79
```

Σχολιασμός: Υλοποιήσαμε Multinomial Naive Bayes Classifier (με μοντέλο εγγράφων διάνυσμα με term occurrences) και Bernoulli Naive Bayes (με μοντέλο εγγράφων με δυαδικές εμφανίσεις όρων). Στη πρώτη περίπτωση, μετά και τα βήματα προεπεξεργασίας που έγιναν, πετύχαμε accuracy 80%, ενώ στη δεύτερη 79%.

Σύγκριση αποτελεσμάτων

Συγκρίνοντας τους δύο τρόπους υλοποίησης, με το λογισμικό Rapid Miner και τον κώδικα που γράψαμε σε Python, παρατηρούμε πως η χρήση λογισμικού πέτυχε μικρότερο ποσοστό ακρίβειας και στα δύο μοντέλα (binary term occurrences και term occurrences). Συγκεκριμένα, βλέπουμε για το πρώτο μοντέλο μια διαφορά της τάξης περίπου του 5% (75.3% και 80%), ενώ για το δεύτερο μοντέλο μια διαφορά περίπου στο 6% (73.15% και 79%).