Χαροκόπειο Πανεπιστήμιο
Τμήμα Πληροφορικής και Τηλεματικής

Συστήματα Διαχείρισης Δεδομένων Μεγάλης Κλίμακας
Άσκηση

Διδάσκων **Δημήτρης Μιχαήλ**
Ακ. Έτος **2019-2020**

Οδηγίες Παράδοσης

1. Η άσκηση μπορεί να γίνει σε ομάδες έως και 3 φοιτητών.
2. Η παράδοση της άσκησης πρέπει να γίνει ηλεκτρονικά μέσω της πλατφόρμας `http://eclass.hua.gr`. Μπορείτε να ανεβάσετε την άσκηση σας μέχρι και την ημέρα της παράδοσης.
3. Το παραπάνω zip αρχείο πρέπει να περιέχει
   (a) ένα φάκελο **src** με τον πηγαίο κώδικα της άσκησης
   (b) ένα .pdf αρχείο με την αναφορά.

   Το αρχείο πρέπει να περιέχει μόνο τον πηγαίο κώδικα και όχι και τα εκτελέσιμα αρχεία.
4. Η αναφορά πρέπει να περιέχει εισαγωγή στο θέμα, λεπτομερή ανάλυση της λύσης που υλοποιήσατε μαζί με τον κώδικα που γράψατε καθώς και παραδείγματα εκτέλεσης του.
5. Σε περίπτωση αντιγραφής θα μηδενίζονται **όλες** οι εμπλεκόμενες ασκήσεις.

Άσκηση

Στην άσκηση αυτή καλείστε να υλοποιήσετε τον αλγόριθμο TextRank για την εξαγωγή λέξεων κλειδιών σε κείμενα γραμμένα στα αγγλικά. Ο συγκεκριμένος αλγόριθμος είναι μία πολύ κλασική τεχνική που πρωτοπαρουσιάστηκε με την παρακάτω δημοσίευση:

- Mihalcea, R., & Tarau, P. (2004, July). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing (pp. 404-411).

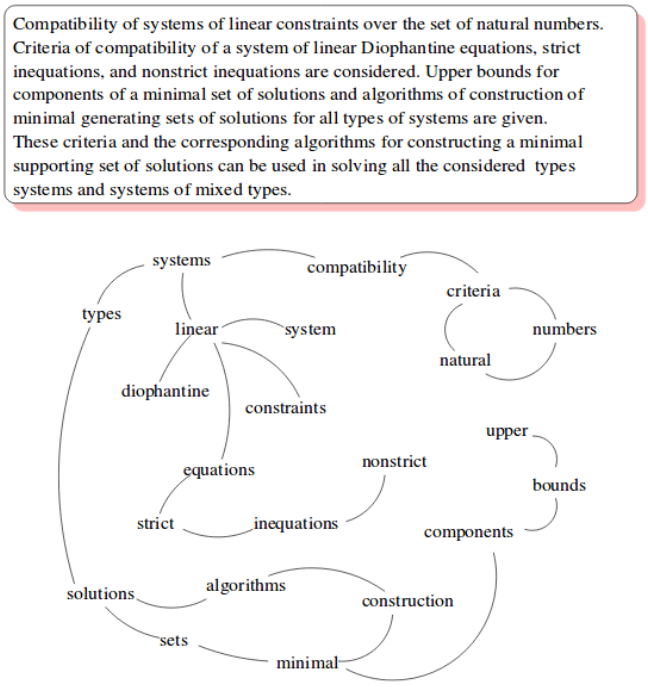
Η συγκεκριμένη δημοσίευση είναι διαθέσιμη εδώ (`https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf`). Διαβάστε προσεκτικά το κείμενο αυτό τουλάχιστον μέχρι και το σημείο όπου συζητάει την εξαγωγή λέξεων κλειδιών.

Κεντρική Ιδέα

Αρχικά ο αλγόριθμος διαβάζει ένα κείμενο και το μετατρέπει σε γράφημα. Οι κόμβοι του γραφήματος είναι οι λέξεις του κειμένου. Ακμές μεταξύ των λέξεων προσθέτονται εαν εμφανίζονται μαζί σε ένα μικρό κινούμενο παράθυρο N λέξεων. Το N είναι υπερπαράμετρος και είναι συνήθως μεταξύ 3 και 10. Αφού το κείμενο μετατραπεί σε ένα γράφημα, τρέχουμε τον αλγόριθμο PageRank και αναθέτουμε έναν αριθμό σε κάθε κόμβο και άρα σε κάθε λέξη του γραφήματος. Για να βγάλουμε τις πιο σημαντικές λέξεις, ταξινομούμε τις λέξεις με βάση το PageRank που μόλις υπολογίσαμε σε μη-αύξουσα σειρά και κρατάμε τις top-k όπου πάλι το k είναι υπερπαράμετρος.

Η παραπάνω ιδέα λειτουργεί πολύ καλύτερα αν δεν χρησιμοποιήσουμε όλες τις λέξεις αλλά μόνο μερικά μέρη του λόγου όπως π.χ ουσιαστικά και επίθετα. Αυτό σημαίνει πως πριν φτιάξουμε το γράφημα από το κείμενο χρειάζεται να το περάσουμε από ένα συντακτικό φίλτρο και να κρατήσουμε μόνο ουσιαστικά και επίθετα.

Όσο αφορά το γράφημα μπορούμε να κατασκευάσουμε κατευθυνόμενο ή μη-κατευθυνόμενο γράφημα. Ανάλογα με την υλοποίηση του PageRank μας, αυτό σημαίνει πως μπορεί να χρειαστεί να προσθέσουμε και ανάδρομες ακμές.



Τεχνολογιές

Για την υλοποίηση σας είναι υποχρεωτικό να χρησιμοποιήσετε το σύστημα Spark.

- Προσοχή, η βαθμολογία σας θα εξαρτηθεί σε σημαντικό βαθμό στο ποσοστό του κώδικα σας που θα τρέχει στο κατανεμημένο σύστημα σε αντίθεση με το ποσοστό που θα τρέχει τοπικά στον driver.
- Επιτρέπεται να χρησιμοποιήσετε είτε Java είτε Python αλλά όχι άλλες γλώσσες προγραμματισμού.
- Προσοχή για να γίνει δεκτή η άσκηση σας θα πρέπει να χρησιμοποιεί το Spark σε επίπεδο RDD και όχι να χρησιμοποιεί κάποια πιο υψηλού επιπέδου υλοποίηση όπως π.χ τα Mllib ή GraphX. Με άλλα λόγια, δεν επιτρέπεται να χρησιμοποιήσετε βιβλιοθήκη του Spark πλέον της κύριας (core).
- Για την υλοποίηση του συντακτικού φίλτρου μπορείτε να χρησιμοποιήσετε έτοιμη βιβλιοθήκη. Για παράδειγμα σε Java την NLP ή για Python την NLTK.
- Για τον PageRank κάναμε μία πολύ απλή υλοποίηση στο εργαστήριο την οποία και μπορείτε αν θέλετε να χρησιμοποιήσετε.

Dataset

Θα πρέπει να εκτελέσετε τον αλγόριθμο σας στο dataset **Inspec** που αποτελείται από 2000 μικρά abstracts. Για κάθε αρχείο εισόδου θα πρέπει να υπολογίσετε τους top-k όρους. Το dataset μπορείτε να το κατεβάσετε από εδώ (`https://github.com/LIAAD/KeywordExtractor-Datasets/raw/master/datasets/Inspec.zip`).

Είσοδος-Έξοδος και Υπερπαράμετροι

Εκτός από τις παραμέτρους για είσοδο και έξοδο θα πρέπει να αφήσετε ως είσοδο από τον χρήστη το $3 \leq N \leq 10$, το $1 \leq k \leq 10$ και τον αριθμό τον iterations του PageRank.

Βαθμολογία

Κριτήρια

- Καλή μοντελοποίηση, χρήση λίγων RDD και σωστό caching των ενδιάμεσων αποτελεσμάτων.
- Σωστή ονοματολογία μεταβλητών και συναρτήσεων.
- Σωστή λειτουργικότητα.
- Αποδοτική υλοποίηση.
- Εύκολη μεταγλώττιση και εκτέλεση.
- Ολοκληρωμένη και σωστή τεκμηρίωση και περιγραφή στην αναφορά.

Μπορειτε να χρησιμοποιήστε (a) Java και Maven όπως στις εργαστηριακές ασκήσεις ή (b) Python. Μπορείτε να χρησιμοποιήσετε ως σκελετό τον κώδικα του εργαστηρίου. Σε περίπτωση Python θα πρέπει να είναι τουλάχιστον έκδοση 3 και να παρέχεται και ένα requirements.txt με όλα τα dependencies.

Εκτός από τα αρχεία με τον κώδικα πρέπει να γράψετε και μια αναφορά. Η αναφορά πρέπει να εξηγεί τις διάφορες επιλογές που κάνατε, γιατί μοντελοποιήσατε έτσι το πρόβλημα καθώς και να σχολιάζει τον κώδικα σας. Η αναφορά πρέπει να είναι υποχρεωτικά σε μορφή *pdf*. Επίσης φροντίστε ο κώδικας σας να περιέχει και ένα README αρχείο που να εξηγεί με ακρίβεια πως κάνει κάποιος compile και πως το εκτελεί. Ακρίβεια εδώ σημαίνει πως απλά κάποιος κάνει copy-paste μία εντολη και την εκτελεί χωρίς να είναι απαραίτητη κάποια αλλαγή.