Ανάπτυξη Λογισμικού για Πληροφοριακά Συστήματα
Χειμερινό εξάμηνο 2017-2018

"N-grams detection"

Ημερομηνία Παράδοσης Α' Μέρους: **03 Νοεμβρίου 2017**

Υπεύθυνος Καθηγητής	**Συνεργάτες μαθήματος**
Ιωαννίδης Ιωάννης	Γαλούνη Κωνσταντίνα
	Θεοδωρακόπουλος Ευθύμιος
	Λιβισιανός Τσαμπίκος
	Πασκαλής Σαράντης

Γενική Περιγραφή

Η εργασία του μαθήματος “Ανάπτυξη Λογισμικού για Πληροφοριακά Συστήματα” βασίζεται στον διαγωνισμό που πραγματοποιήθηκε στο Sigmod το έτος 2017. Για περισσότερες πληροφορίες ο χρήστης μπορεί να ανατρέξει στο σύνδεσμο για το διαγωνισμό του συνεδρίου.
(http://sigmod17contest.athenarc.gr/).

N-grams

Τα N-grams χρησιμοποιούνται συνήθως στην επεξεργασία φυσικής γλώσσας και στην εξαγωγή πληροφοριών. Ένα N-gram είναι μια ακολουθία Ν λέξεων. Για παράδειγμα, αν έχουμε την φράση “the book is on the table” και θέλουμε να εξάγουμε όλα τα N-grams με N=3 τότε θα έχουμε τα εξής:

- the book is
- book is on
- is on the
- on the table

Ανάλυση task

Στόχος της εργασίας είναι ο εντοπισμός των N-grams σε κείμενα. Συγκεκριμένα, έχουμε μια δεδομένη λίστα από N-grams (με διάφορα Ν), την οποία κρατάμε στην μνήμη και μια ροή από κείμενα στα οποία πρέπει να εντοπίσουμε πιθανές εμφανίσεις των N-grams. Κατά την ροή των κειμένων θα δίνονται εντολές προσθήκης/αφαίρεσης N-grams οι οποίες θα ανανεώνουν την αρχική λίστα.

Περιγραφή παραδοτέων μαθήματος

Η εργασία θα χωριστεί σε τρία επίπεδα. Το κάθε επίπεδο θα υλοποιηθεί σε συγκεκριμένη χρονική περίοδο και θα βασίζεται στα προηγούμενα επίπεδα. Ο διαχωρισμός των επιπέδων στοχεύει: στην ανάπτυξη μιας απλής λύσης του προβλήματος και έπειτα στη σταδιακή βελτιστοποίηση των επιμέρους τμημάτων, βελτιώνοντας με αυτό τρόπο την αποδοτικότητα της εφαρμογής.

Στη συνέχεια παρουσιάζονται οι λειτουργίες που θα αναπτυχθούν στο πρώτο επίπεδο.

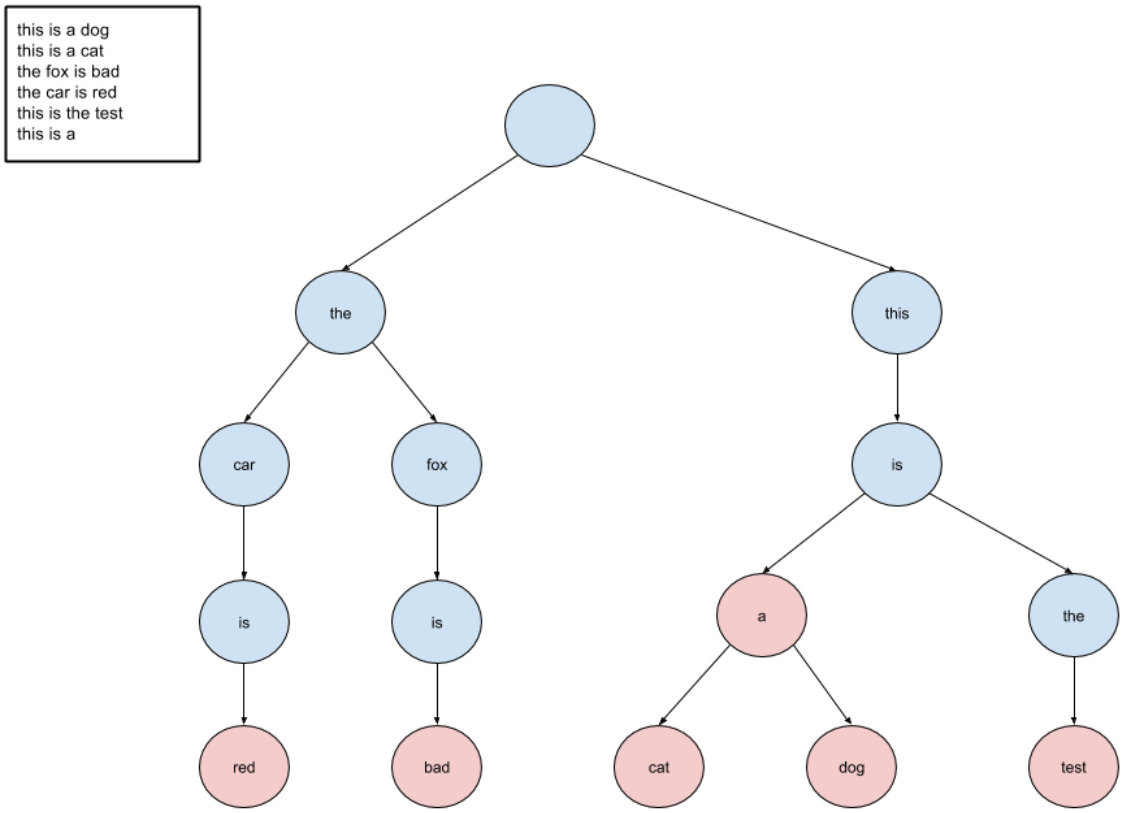
1. Είσοδος και αποθήκευση στο trie
2. Υλοποίηση αλγορίθμου επίλυσης του προβλήματος
3. Υλοποίηση test για τον έλεγχο των δομών δεδομένων και της αναζήτησης

Α' μέρος της Άσκησης

Για την υλοποίηση του συγκεκριμένου επιπέδου της εργασίας, θα χρησιμοποιηθεί η δομή του trie, με μια μικρή παραλλαγή, για την αποθήκευση των N-grams. Πιο αναλυτικά:

1. Είσοδος και αποθήκευση στο trie

Αρχικά το πρόγραμμα σας θα πρέπει να διαβάσει το αρχείο με τα N-grams και να τροφοδοτήσει το αρχικό trie. Στο παρακάτω σχήμα φαίνεται η μορφή που θα πρέπει να έχει μετά από εισαγωγή των 6 N-grams του πλαισίου.



Το trie κρατάει τα παιδιά ενός κόμβου ταξινομημένα αλφαριθμητικά. Αυτό σημαίνει ότι δεν εξαρτάται από την σειρά με την οποία έρχονται τα N-grams στην είσοδο. Για παράδειγμα, όταν

έρθει το “the” θα μπει στα αριστερά του “this” ώστε να διατηρηθεί η ταξινόμηση. Αντίστοιχα, το “car” θα μπει στα αριστερά του “fox”.

Επιπλέον στο σχήμα όταν ένας κόμβος αναπαριστάται με κόκκινο χρώμα, περιέχει λέξη η οποία είναι τελική για κάποιο N-gram (όπου N στη συγκεκριμένη περίπτωση είναι το βάθος του κόμβου-1, αφού ο κόμβος-ρίζα δεν περιέχει κάποια λέξη)

Δομή αναπαράστασης trie

Για την αναπαράσταση του trie δίνεται **ενδεικτικά** τα παρακάτω struct.

```
struct index{
    trie_node* root;
    ...
};

struct trie_node{
    char* word;
    trie_node** children;
    char is_final;
    ...
};
```

Για τα παιδιά κάθε κόμβου θα δεσμεύεται αρχικά ένας πίνακα N μεγέθους. Καθώς το πλήθος των παιδιών ενός κόμβου δεν είναι γνωστό από την αρχή, σε περίπτωση που ο πίνακας γεμίσει θα πρέπει να διπλασιάζεται ώστε να χωρέσει τα υπόλοιπα.

2. Υλοποίηση αλγορίθμου επίλυσης του προβλήματος

Μετά την αποθήκευση του trie, στην είσοδο της εφαρμογής σας θα δίνονται ριπές εργασιών. Κάθε ριπή περιλαμβάνει μία σειρά εργασιών (μια εργασία ανά γραμμή). Το τέλος της ριπής το σηματοδοτεί μια γραμμή που περιέχει μόνο το χαρακτήρα “F”.

Κάθε εργασία θα πρέπει να εκτελείται χωρίς να την επηρεάζουν οι επόμενες εργασίες της ριπής που ανήκει. Τα αποτελέσματα των ερωτημάτων που περιέχονται μέσα σε μια ριπή πρέπει να υπολογίζονται και να εμφανίζονται πριν την εκτέλεση των εργασιών της επόμενης ριπής. Μετά το τέλος κάθε πακέτου πράξεων, αναμένεται να εμφανίσετε στην έξοδο της εφαρμογής σας το αποτέλεσμα των εργασιών, πριν διαβάσετε την επόμενη ριπή.

Κάθε εργασία αναπαρίσταται από έναν χαρακτήρα (“Q” ή “A” ή “D”) ο οποίος ορίζει και τον τύπο της εργασίας.

Οι τρεις τύποι εργασιών είναι:

“Q”/ερώτημα: Αυτή η εργασία θέτει ένα ερώτημα εύρεσης N-gram σε ένα κείμενο Για παράδειγμα, το ερώτημα “Q this is a cat and a dog” στο παραπάνω trie του σχήματος, θα πρέπει να μας επιστρέψει τα N-grams “this is a cat” και “this is a”. Η απάντηση για κάθε Q θα πρέπει να δίνεται σε μια γραμμή και κάθε N-gram θα πρέπει να χωρίζεται με τον χαρακτήρα “|”. Η σειρά με την οποία θα πρέπει να δίνονται τα N-grams είναι με τη σειρά που βρίσκονται στο κείμενο, άρα για το παραπάνω ερώτημα η σωστή απάντηση είναι “this is a|this is a cat”. Σε περίπτωση που μέσα στο κείμενο δεν υπάρχει κάποιο από τα N-grams τότε η απάντηση είναι -1.

“A”/προσθήκη: Αυτή η εργασία απαιτεί από τον χρήστη να τροποποιήσει το trie προσθέτοντας ένα ακόμη N-gram. Θα πρέπει και εδώ να δοθεί προσοχή ώστε το καινούργιο N-gram να μπει στη σωστή θέση στο trie. Σε περίπτωση που το N-gram υπάρχει ήδη το trie παραμένει αναλλοίωτο. Η συγκεκριμένη εργασία δεν παράγει κάποια έξοδο.

“D”/διαγραφή: Αυτή η εργασία απαιτεί από τον χρήστη να τροποποιήσει το trie αφαιρώντας από το trie ένα N-gram. Αν το συγκεκριμένο N-gram δεν υπάρχει τότε το trie παραμένει αναλλοίωτο. Η συγκεκριμένη εργασία δεν παράγει κάποια έξοδο.

Ακολουθεί ένα παράδειγμα από μια ριπή εργασιών και της αναμενόμενης εξόδου δεδομένου του παραπάνω trie:

Q this is a cat and a dog Q this is not the test A test Q this is not the test A fast car D the car is red Q the car is red and fast F	this is a|this is a cat -1 test -1

Η ριπή στο παράδειγμα έχει 4 ερωτήματα άρα θα πρέπει να επιστρέψει 4 γραμμές στην έξοδο. Για δοκιμές μπορείτε να χρησιμοποιήσετε το εργαλείο στη σελίδα του διαγωνισμού (http://sigmod17contest.athenarc.gr/task.shtml).

Ενδεικτικά πρότυπα συναρτήσεων

```
trie_node* init_trie( );  // επιστρέφει το root

trie_node* create_trie_node( );
```

```
OK_SUCCESS insert_ngram(index* ind, … )

OK_SUCCESS delete_ngram(index* ind, … )

char* search(index* ind, …)

...
```

Τα πρότυπα είναι ενδεικτικά. Επιπλέον πρότυπα μπορούν να προστεθούν, καθώς και επιπλέον ορίσματα στα ήδη υπάρχοντα.

3. Υλοποίηση test για τον έλεγχο των δομών δεδομένων και της αναζήτησης

Θα πρέπει να φτιάξετε τρόπους ελέγχου των βασικών λειτουργιών που αναπτύξατε για τη δομή αποθήκευσης του trie και για την αναζήτηση των N-gram. Ουσιαστικά, θα πρέπει για συγκεκριμένη είσοδο το πρόγραμμα να παράγει την αναμενόμενη έξοδο. Καλό είναι το πρόγραμμα σας να προκαλεί δυναμικές μεταβολές στις δομές σας (π.χ. διπλασιασμός) προκειμένου να εξασφαλίσετε ότι λειτουργούν σωστά. Ένας προτεινόμενος τρόπος για να ελέγχετε τις δομές σας είναι με τη χρήση unit tests (https://en.wikipedia.org/wiki/Unit_testing).

Παρατηρήσεις

- Μπορείτε να χρησιμοποιήσετε είτε c είτε c++. Στην περίπτωση της c++ απαγορεύεται η χρήση των βιβλιοθηκών της stl.
- Θα πρέπει να δοθεί makefile
- Η εκτέλεση του προγράμματος θα γίνεται: ./ngrams -i <init_file> -q <query_file> και η εκτύπωση θα γίνεται στο stdout
- Μαζί θα πρέπει να υπάρχει εκτενής αναφορά που να εξηγεί τις επιλογές σας
- Στο εβδομαδιαίο μάθημα, κρίνεται υποχρεωτική η παρουσία τουλάχιστον ενός μέλους της ομάδας