EFFICIENT TOP-K QUERYING OVER SOCIAL-TAGGING NETWORKS

Ralf Schenkel, Tom Crecelious, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane Xavier Parreira, Gerhard Weikum

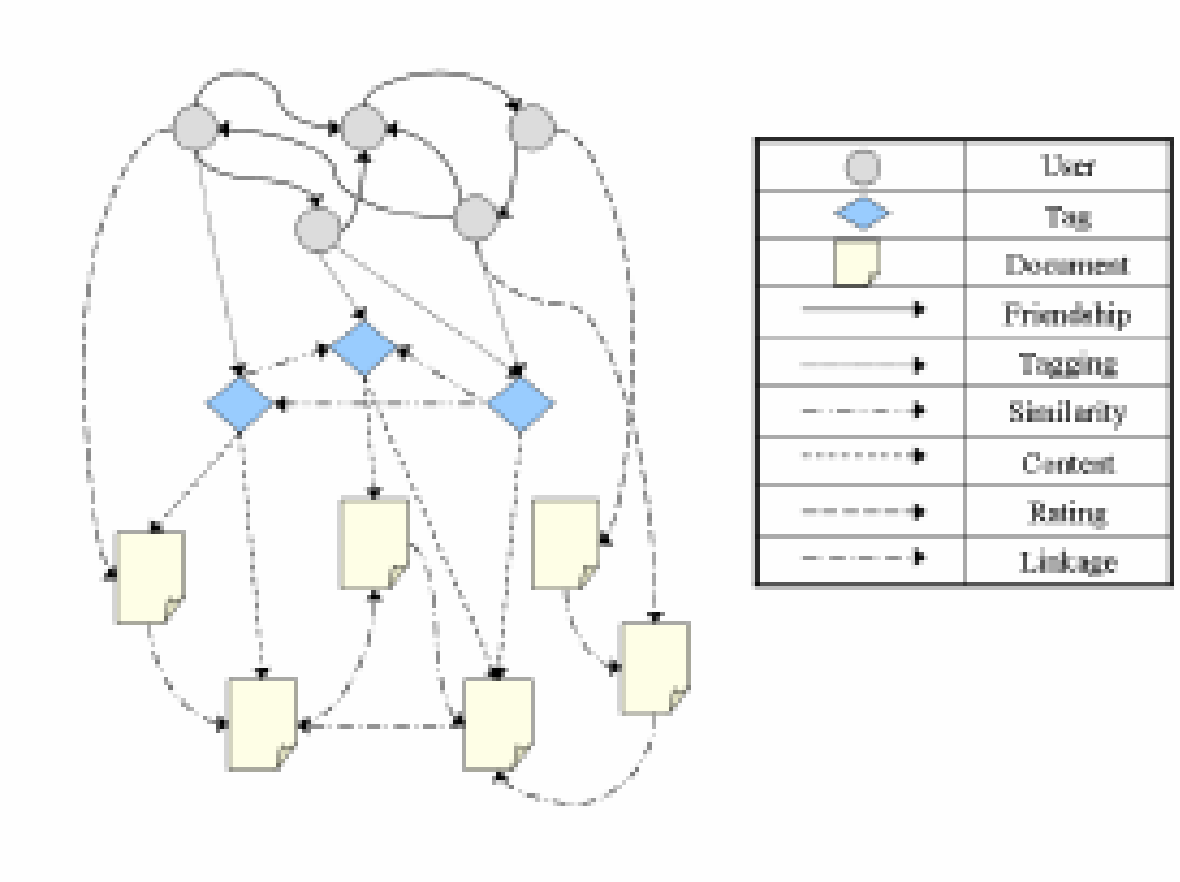
ΠΡΟΒΛΗΜΑ

- Εύρεση ενός αποτελεσματικού αλγορίθμου top-k σε Social-tagging networks λαμβάνοντας υπόψη τα εξής:
  - την σχέση μεταξύ των χρηστών - social expansion
  - την ομοιότητα των ετικετών (tags) - semantic expansion
- Λύση του προβλήματος
  - Χρήστη ενός αλγορίθμου top-k συγκεκριμένα τον threshold algorithm (TA) που λαμβάνει υπόψη του τα παραπάνω στοιχεία που αναφέραμε

SOCIAL-TAGGING NETWORK SOCIETY

- Σύνολο χρηστών
- Σχέσεις φιλίας μεταξύ των χρηστών
- Σύνολο αντικειμένων (βιβλία, μουσική, βίντεο, φωτογραφίες…)
- Ετικέτες ή βαθμολογίας στα αντικείμενα της αρεσκείας μας.

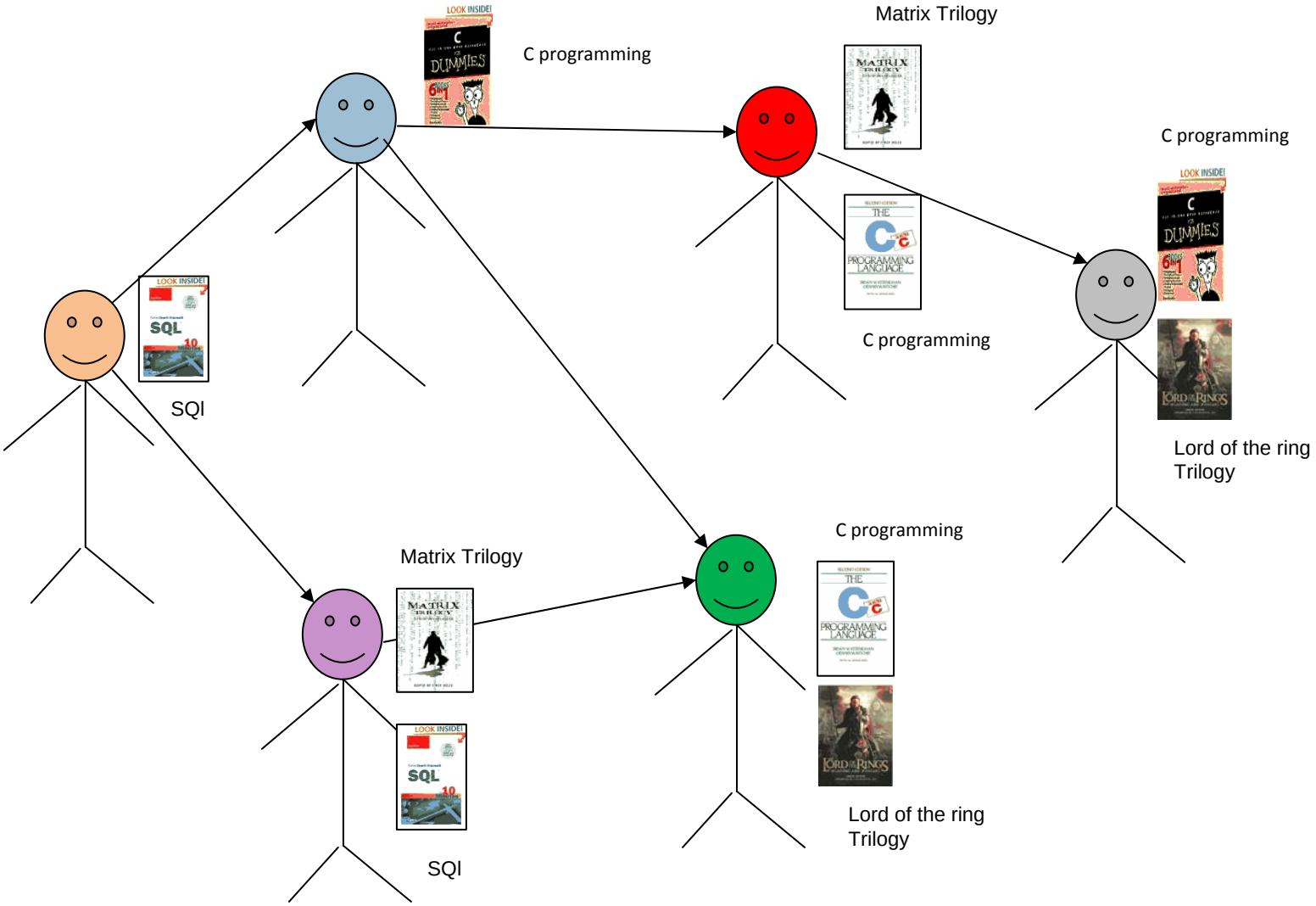
SOCIAL NETWORK MODEL



ΣΧΕΣΕΙΣ ΜΕΤΑΞΥ ΤΩΝ ΚΟΜΒΩΝ ΤΟΥ ΓΡΑΦΟΥ

- Friendship(User1, User2, Friendship Strength)
- TagSimilarity(Tag1, Tag2, Tag Sim)
- Linkage(Document1, Document2, Weight)
- DocContent(Document, Tag, Content Score)
- Tagging(User, Tag, Tag Score)
- Rating(User, Document, Rating Score)

SOCIAL NETWORK MODEL



ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΣΚΟΡ

- Για τον υπολογισμό του σκορ λαμβάνουμε υπόψη μας τα εξής
  - Την σημαντικότητα τον χρηστών, που έχουν κάποια σχέση με τον χρήστη που κάνει την ερώτηση
  - Την συχνότητα με την οποία οι χρήστες μπορεί να χρησιμοποιούν μια ετικέτα.
  - Και την ομοιότητα που μπορεί να υπάρχει μεταξύ των ετικετών
- Συμβολισμοί
  - Q(u,q1…qn) ερώτηση
  - U σύνολο χρηστών
  - D σύνολο εγγράφων
  - T σύνολο ετικετών

FRIENDSHIP SIMILARITY

- Επικαλυπτόμενη ομοιότητα για τους χρήστες που συνδέονται απευθείας μεταξύ τους

$$O(u,u') = \frac{2 * |tagset(u \cap u')|}{|tagset(u)| + |tagset(u')|}$$

- Ομοιότητα για τους χρήστες που δεν συνδέονται απευθείας

$$Pu(u') = \max \prod_{i=0}^{k-1} O(u_i, u_{i+1})$$

- Τέλος η friendship similarity

$$F_u(u') = \alpha * \frac{1}{|U|} + (1-\alpha) P_u(u')$$

SOCIAL FREQUENCY

$$sf_u(d,t) = \sum_{u' \in U} F_u(u') * tf_{u'}(d,t)$$

- Προσθέτουμε τον όρο $tf_u(d,t)$ που είναι το πλήθος των φορών ο χρήστης u έχει χρησιμοποιήσει την ετικέτα t για το έγγραφο d.

$$sf_u(d,t) = \sum_{u' \in U} \left(\frac{a}{|U|} * tf_{u'}(d,t) + (1-a) * F_u(u') * tf_{u'}(d,t) \right)$$

global part
$TF(d,t)$

user spesific part

ΟΜΟΙΟΤΗΤΑ (SIMILARITY)

- Χρήση του τύπου Okapi (BM25)

$$s_u(d,t) = \frac{(k_1+1) * |U| * sf_u(d,t)}{k_1 + |U| * sf_u(d,t)} * idf(t)$$

- Όπου *idf(t)* the inverse document frequency of tag

$$idf(t) = \log \frac{|D| - df(t) + 0.5}{df(t) + 0.5}$$

- Ομοιότητα μεταξύ δυο ετικετών

$$tsim(t,t') = P[t'|t] = \frac{df(t \cap t')}{df(t)}$$

- Ομοιότητα εγγράφου με tag expansion

$$s'_u(d,t) = \max_{t' \in T} tsim(t,t') * s_u(d,t')$$

ΑΛΓΟΡΙΘΜΟΣ

- Κρατάμε τέσσερεις ταξινομημένες λίστες
  - DOCS(t): Περιέχει το πλήθος τον φορών που οι χρήστες έχουν βάλει την ετικέτα t στο document d (έχουμε την τιμή του TF(d,t) σε φθίνουσα σειρά).
  - USERDOCS(u, t): Περιέχει το σύνολο τον documents d που έχουν σημειωθεί από τον χρήστη u με μια ετικέτα t και την αντίστοιχη τιμή $tf_u(d,t)$ (που συνήθως έχει την τιμή 1).
  - FRIENDS(u): Περιέχει το σύνολο τον φίλων του χρήστη u και την ομοιότητα μεταξύ τους ταξινομημένη σε φθίνουσα σειρά.
  - SIMTAGS(t): Περιέχει για την ετικέτα t όλες τις παρόμοιες ετικέτες t' μαζί με την ομοιότητα τους ταξινομημένη σε φθίνουσα σειρά.

ΨΕΥΔΟΚΩΔΙΚΑΣ ΑΛΓΟΡΙΘΜΟΥ

Procedure CONTEXTMERGE(user u,query $q_1 \ldots q_n$,a)
    For i= 1…n do
        FRIENDS[i]=FRIENDS(u);
        DOCS[i]=DOCS(qi);
    End for
    Candidates=0;
    Repeat
        For b=1…batchsize **do**
            L=CHOOSENEXTLIST();
            If l=FRIENDS[i] **then**
                Read USERDOCS(FRIENDS[i],q[i])
                Go to next entry in FRIENDS[i];
            Else if l=DOCS[i] **then**
                Read DOCS[i];
            End if
        End for
        CHECKRANDOMACCESSES();
        If CHECKTERMINATION() **then**
            Break;
        End if
    Until termination
End procedure

ΕΠΕΞΗΓΗΣΗ ΑΛΓΟΡΙΘΜΟΥ (1)

- Για την ερώτηση Q ο αλγόριθμος τρέχει διαδοχικά για κάθε tag και σαρώνει τις λίστες DOCS και USERDOCS για να υπολογίσει το υψηλότερο σκορ.
- Η λίστα DOCS[i] συνεισφέρει στο σκορ $s_u(d,t)$ βάζοντας το $TF(d,t)=high[i]$ και θέτοντας το user specific part 0 (όπου high η μεγαλύτερη τιμή από την λίστα DOCS[i])
- Η Friend[i] λίστα κρατάει την υψηλότερη τιμή της $highF[i]$ και συνεισφέρει στο σκορ $s_u(d, t)$ με την ποσότητα:

$$\frac{(k_1+1)*((1-\alpha)|U|*highF[i]*maxtf(q_i))}{k_1+((1-\alpha)|U|*highF[i]*maxtf(q_i))}*idf(t)$$

όπου $maxtf(q_i)$ η μέγιστη εμφάνιση της ετικέτας για οποιονδήποτε χρήστη.

ΕΠΕΞΗΓΗΣΗ ΑΛΓΟΡΙΘΜΟΥ (2)

- Κατά την διάρκεια της εκτέλεσης κρατάμε δυο λίστες:
  - Στην μία κρατάμε τα top-k μέχρι στιγμής έγγραφα
  - Στην άλλη τα έγγραφα που είναι υποψήφια για να μπουν στα στην τελική λίστα με τα top-k έγγραφα.
- Έτσι για κάθε έγγραφο υπολογίζουμε ένα άνω και κάτω όριο του σκορ $s_u(d,t)$

ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΑΝΩ ΚΑΙ ΚΑΤΩ ΟΡΙΟΥ ΤΟΥ ΣΚΟΡ

- Κάτω όριο του σκορ $s_u(d_j, t)$ υπολογίζεται θέτοντας:
  - $TF(d_j, q_i)$ μηδέν για τα ι για τα οποία δεν έχουμε δει ακόμα το έγγραφο dj.
  - την ποσότητα $\sum_{u' \notin U} R_{u'}(u') * tf_{u'}(d_j, q_i)$ με $uf(d_j, q_i) = \sum_{USERDOCS(u') read\ for\ qi} R_u(u') * tf_{u'}(d_j, q_i)$

- Άνω όριο του $s_u(d, t)$ υπολογίζεται θέτοντας:
  - $TF(d_j, q_i)=high[i]$
  - και το user specific part με $uf(d_j, q_i) + C$ όπου C η συνεισφορά στο σκορ των χρηστών που δεν έχουμε δει ακόμα και δίνεται από τον τύπο:

$$C = \left(1 - \sum_{USERDOCS(u') read\ for\ qi} R_u(u')\right) * maxtf$$

ΤΕΡΜΑΤΙΣΜΟΣ ΑΛΓΟΡΙΘΜΟΥ

- Για τον υπολογισμό των top-k εγγράφων αλγορίθμου συγκρίνουμε το ελάχιστο κάτω φράγμα (min-k) των εγγράφων που βρίσκονται στην current top-k λίστα με το μέγιστο άνω φράγμα των εγγράφων που από την λίστα των υποψηφίων και
  - αν είναι μεγαλύτερο τότε έχουμε βρει τα top-k έγγραφα
  - αλλιώς ανανεώνουμε την λίστα προσθέτοντας το υποψήφιο έγγραφο.

TAG-EXPANSION

- Στον υπολογίσουμε το άνω και το κάτω σκορ προστίθεται η ποσότητα $tsim(q_i,t_{ij})$.
- Κρατάμε για κάθε ετικέτα q_i την αντίστοιχη λίστα SIMTAGS(q_i) που έχει ταξινομημένη την ομοιότητα $tsim(q_i,t_{ij})$ σε φθίνουσα σειρά
- Επιλογή της λίστας όπως γίνεται και για τις λίστες DOCS[i] και FRIENDS[i], αφού επιλεγεί η λίστα διαβάζουμε τις αντίστοιχες λίστες για την συγκεκριμένη ετικέτα.

DATA SET

- Del.icio.us (http://del.icio.us): Με 12,389 χρήστες, 2,781,096 ετικέτες, 152,306 φιλικές συνδέσεις μεταξύ των χρηστών.
- Flickr (http://flickr.com/): Με 52,347 χρήστες, 29,111,183 ετικέτες, 1,293,777φιλικές συνδέσεις μεταξύ των χρηστών.
- LibraryThing (http://librarything.com): Με 9,986 χρήστες, 14,296,693 ετικέτες, 17,317 φιλικές συνδέσεις μεταξύ των χρηστών.

ΠΕΙΡΑΜΑ 1

- User-specific ground truth
  - Delicious 150 ερωτήσεις
  - Library Thing 184 ερωτήσεις
- User study
  - Librarything 28 ερωτήσεις
  - Flickr 40 ερωτήσεις
- Precision

Without Tag Expansion											
α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Delicious	0.15	0.25	0.27	0.33	0.34	0.35	0.35	0.37	0.39	0.39	0.36
LibraryThing	0.29	0.42	0.49	0.54	0.53	0.55	0.56	0.59	0.60	0.63	0.62
With Tag Expansion											
α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Delicious	0.16	0.25	0.28	0.36	0.36	0.36	0.36	0.39	0.40	0.37	0.36
LibraryThing	0.30	0.41	0.46	0.53	0.52	0.53	0.55	0.57	0.59	0.61	0.60

- NDCG

Without Tag Expansion											
α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Flickr	0.39	0.42	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.36
LibraryThing	0.61	0.65	0.65	0.66	0.67	0.66	0.66	0.68	0.70	0.72	0.71
With Tag Expansion											
α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Flickr	0.42	0.40	0.40	0.40	0.40	0.39	0.39	0.40	0.40	0.40	0.36
LibraryThing	0.61	0.63	0.64	0.65	0.65	0.65	0.65	0.67	0.69	0.72	0.71

ΠΕΙΡΑΜΑ 2

- Μέσο κόστος εκτέλεσης

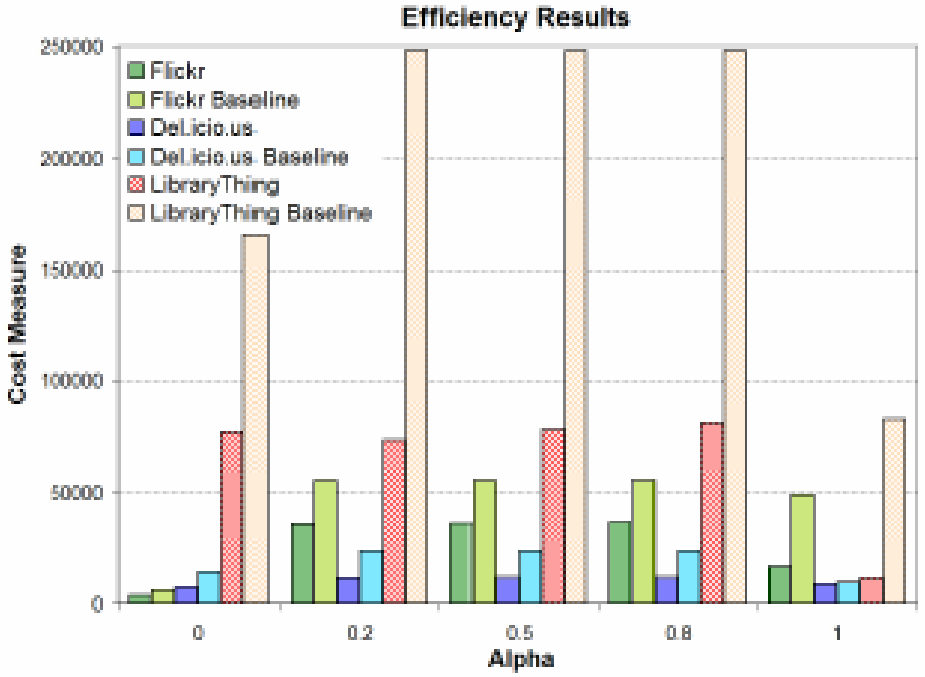


Figure 3: Average Execution Cost without tag expansion

ΠΕΙΡΑΜΑ 2 (TAG EXPANSION)

- Μέσο κόστος εκτέλεσης

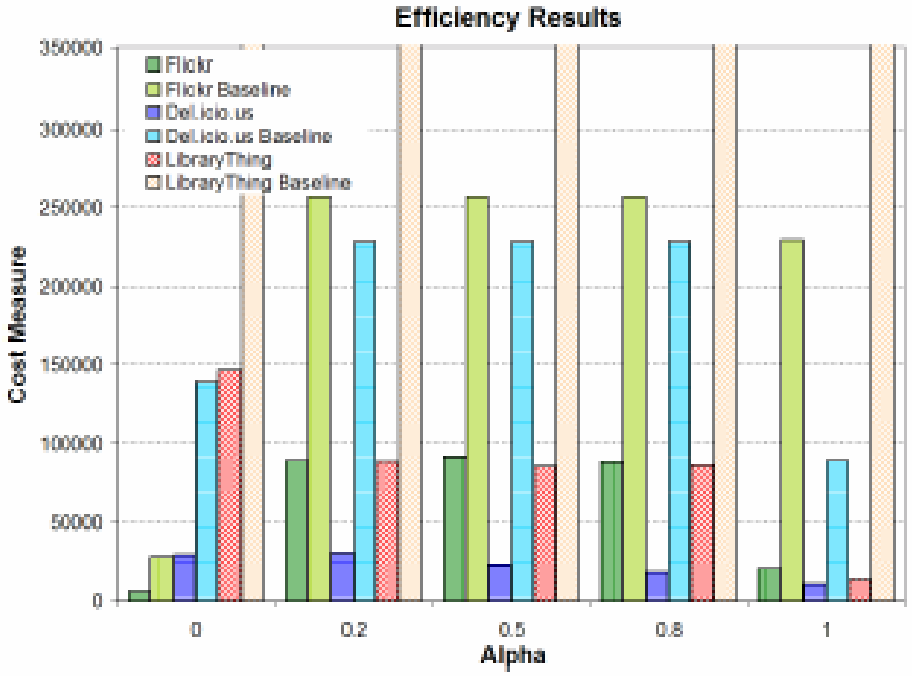


Figure 4: Average Execution Cost with tag expansion

ΣΥΜΠΕΡΑΣΜΑΤΑ

- Ο νέος αλγόριθμος για τον υπολογισμό των top-k εγγράφων σε social-networks λαμβάνει υπόψη του τις σχέσεις που υπάρχουν μεταξύ των χρηστών
- Υπολογίζει το σκορ σταδιακά καθώς εκτελείτε ο αλγόριθμος και όχι από την αρχή.
- Δεν χρειάζεται να υπολογίσει το σκορ για όλα τα έγγραφα
- Πολύ πιο αποτελεσματικός από τον baseline αλγόριθμο

ΕΥΧΑΡΙΣΤΩ

ΕΡΩΤΗΣΕΙΣ;

ΕΡΩΤΗΣΗ

- Τι διαφορά προκύπτει στον αλγόριθμο όταν η μεταβλητή a λαμβάνει τις ακραίες τιμές της (μηδέν ή ένα).