Diversity-aware continuous top-k queries in social networks

Abdulhafiz Alkhouli, Dan Vodislav

▶ **To cite this version:**

Abdulhafiz Alkhouli, Dan Vodislav. Diversity-aware continuous top-k queries in social networks. COOPIS 2017, Oct 2017, Rhodes, Greece. 10.1007/978-3-319-69462-7_7 . hal-01724282

HAL Id: hal-01724282
https://hal.science/hal-01724282v1

Submitted on 6 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Diversity-aware continuous top-k queries in social networks

(short paper)

Abdulhafiz Alkhouli and Dan Vodislav

ETIS - ENSEA / Univ. of Cergy-Pontoise / CNRS - France
{abdulhafiz.alkhouli,dan.vodislav}@ensea.fr

Abstract. We consider here the problem of adding diversity requirements for the results of continuous top-k queries in a large scale social network, while preserving an efficient, continuous processing. We propose the DA-SANTA algorithm, which smoothly adds content diversity to the continuous processing of top-k queries at the social network scale. The experimental study demonstrates the very good properties in terms of effectiveness and efficiency of this algorithm.

Keywords: information streams, social networks, diversity, continuous top-k query processing, publish/subscribe systems

1 Introduction and related work

We consider here the context of *top-k continuous queries* over text information streams produced in social networks. Efficient processing of such queries at the social network scale requires *continuous processing techniques* that incrementally maintain the top-k list of each user in reaction to social network events (new message, interaction with message). However, existing methods have difficulties to handle complex scoring functions, including social network criteria, and usually focus on content-based relevance and time-based factors favoring more recent messages. In previous work [1], we proposed an efficient method for continuous processing of top-k queries over information streams in a large-scale social network, using a relevance model with content-based, time-based and rich social network factors (user- and interaction-based). We extend this work here, to introduce *results diversification* into the continuous processing method.

Result diversification [4] aims at avoiding redundancy and too homogeneous results to often imprecise user queries. For instance, a query about "olympic games" could get only items about the 2024 games abundantly discussed recently, while the user interest may be different. The *content diversity* of a results set is generally measured either by the average or the minimum distance between all the results. The general approach to add diversity to top-k querying is to use a *bi-objective scoring function* that combines relevance and diversity.

Adding results diversification to continuous processing of top-k queries over information streams at a social network scale is a very challenging task. First, the general diversification problem is NP-hard [7]: given a query over a set O of n objects, find $S_k^* \subseteq O$ of size $k \leq n$, which maximizes a bi-objective function that combines relevance and diversity. Specific approximate methods are necessary,

especially in the case of continuously arriving data. For instance, [3] and [5] limit O to a sliding window of the most recent items, while [9] proposes an incremental approach to maintain an approximate diversified top-k set, where O is limited to the new item plus the current top-k. We adopt the latter definition for O, but propose heuristics to select a victim in the current top-k. Next, the constraint of continuous processing at the social network scale requires very efficient algorithms. The methods above are not adapted to large social networks, since they evaluate each new item with all the queries. The only work to date that proposes a diversity-aware method adapted to a large number of queries is [2]. Like us, they index subscription queries to be able to prune queries not affected by a new message, but focus on grouping queries into blocks for efficiency reasons. They also use a victim selection heuristics that considers the oldest message in the top-k. But their relevance function, favoring query grouping, is very specific and they do not consider social network relevance factors.

Finally, when trying to extend an existing relevance-only approach to add diversity, one must face the model mismatch between top-k computation for relevance, at the element level, and diversity, at the set level. Efficient continuous processing of top-k queries at a large scale is based on *index structures for user queries*, with the specificity that they must include information about μ_q, the k-th current score of each query, the limit to overstep to enter the top-k. Most of them use inverted lists, one per query term. In [8], the index list for term t contains subscription queries q containing t, sorted by w_{tq}/μ_q, where w_{tq} is the weight of t in q. In [10] lists are ordered by a threshold value based on the current top-k of each query. Some methods use different structures, for instance [13] employs an original two-dimensional inverted query indexing scheme combining w_{tq} and μ_q. [12] use a double index per query term: an inverted list ordered by query id and a tree structure organized by μ_q. Methods that focus on *grouping strategies* to handle groups of queries instead of individual ones propose specific index structures, for instance [11] uses a graph to index covering relationships between subscription queries. To extend these methods with results diversification, the indexing technique must be flexible enough to support it. This is not the case for most of them, which also explains the fact that they do not consider more complex scoring components, such as social network factors.

In this context, our main contributions are (i) *a model* that smoothly integrates content-based diversity into the continuous top-k processing model presented in [1], including heuristics for approximate diversification and a query indexing structure for efficient processing of diversity-aware top-k queries at the social network scale, and (ii) *an algorithm*, DA-SANTA (Diversity-Aware Social and Action Network Threshold Algorithm), based on this model, whose effectiveness and efficiency are demonstrated through a set of experiments.

2 Data and processing models

Social network information streams. We consider the social network model from [1], with asymmetric relation graphs, where each user produces a single information stream of text messages and issues a single *implicit* subscription query, expressed by the *user profile*. Like messages, user profiles are described by

a set of weighted terms expressing the user's points of interest. The importance of the content of a message m for user u is measured by a similarity function sim (e.g. cosine similarity) between m and u's profile $p(u)$.

The model also considers an asymmetric *importance function* f, where $f(u_1, u_2) \in [0, 1]$ is the importance of user u_2 for u_1 in the social network. Note that, even if f is defined for any couple of users in the network , in practice each user has a limited number of users of interest (with $f > 0$), which results into reasonable effort to manage this information.

Relevance scoring function. We consider the relevance scoring model proposed in [1], combining content-based, time-based and social network factors.

$$tscore(m, u) = score(m, u) \cdot TB(t^m - t_o) \quad (1)$$

$$score(m, u) = a\ sim(m, p(u)) + b\ f(u, u^m) + c\ G(m) \quad (2)$$

(1) gives the time-based relevance of message m for user u. We use a *time bonus function* $TB : \mathbb{R}_+ \rightarrow [1, \infty)$, monotonically increasing, with $TB(0) = 1$, where $t^m \in TS$ is the publishing time for message m and $t_o \in TS$ a fixed origin moment. Time-independent relevance $score(m, u)$ expresses the initial importance of m for u at moment t^m. It combines three elements: (i) *content-based similarity* ($sim(m, p(u))$) between the message and the user profile, (ii) *user-dependent importance of the message* in the social network ($f(u, u^m)$), measured by the importance of the message emitter u^m for user u, and (iii) *user-independent importance of the message* in the social network ($G(m)$), measured at publishing time by the global importance of the emitter in the social network.

Diversity model. We adopt the commonly used *max-sum diversification* bi-criteria objective function [7] to combine relevance and diversity into a single scoring function. If we note $u.TL_k = \{m_1, ..., m_k\}$ the top-k result set for user u, its diversity $D(u.TL_k)$ is given by the sum of distances between the set elements, where $dist(m_i, m_j) = 1 - sim(m_i, m_j)$. The combined relevance-diversity score DR is a linear combination between relevance and diversity.

$$DR(u.TL_k) = \nu\ f_R(u.TL_k) + (1 - \nu)\ f_D(u.TL_k) \quad (3)$$

Here, $f_R(u.TL_k) = \sum_{m \in u.TL_k} rel(m, u)$ expresses the relevance of the top-k list (the rel scoring function may be (1) or (2)), while $f_D(u.TL_k) = \frac{2}{k-1} D(u.TL_k)$ measures the diversity score, where the homogeneity factor $2/(k-1)$ compensates the fact that f_R sums k values, while for f_D we have $k(k-1)/2$ values.

Processing model. We adopt the commonly used approach [9][2] in top-k diversification on streams, to limit the set of objects to the new message plus the current top-k. Hence, for a given user u having the top-k result list $u.TL_k$, when a new message m_{new} arrives, the updated top-k list $u.TL'_k$ will be the subset of size k of $u.TL_k \cup \{m_{new}\}$ that maximizes the relevance-diversity score DR defined in (3). Then the condition for the top-k to be updated is:

$$DR(u.TL'_k) > DR(u.TL_k) \quad (4)$$

The basic algorithm for updating the top-k lists would *repeat the above processing for all the users in the social network*, which raises an important efficiency issue. Also, for each user *the top-k update method is expensive*, since it requires k computations of the DR function.

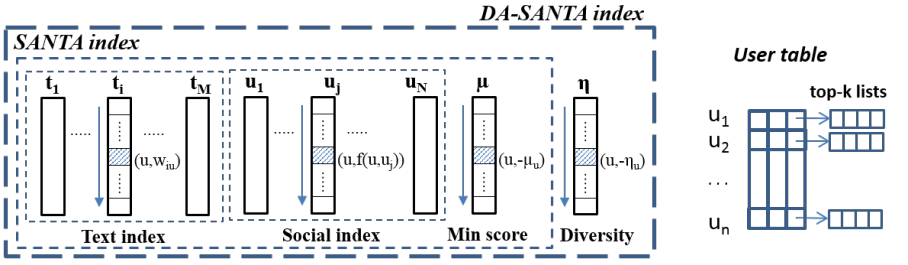


Fig. 1. SANTA and DA-SANTA index and data structures

We propose the DA-SANTA algorithm that provides solutions for both these efficiency problems. DA-SANTA proposes a pruning approach to avoid evaluating all users and employs heuristic methods to choose a single message (victim) to be replaced with the new message.

3 The DA-SANTA algorithm

DA-SANTA scoring. For a new published message m_{new} and a given user u, an heuristic function designates $m_{vic} \in u.TL_k$ as potential victim. As shown in Section 2, the condition for m_{new} to replace m_{vic} in $u.TL_k$ is $DR(u.TL'_k) > DR(u.TL_k)$, where $u.TL'_k = u.TL_k \cup \{m_{new}\}$ - $\{m_{vic}\}$. We note $u.F_k = u.TL_k$ - $\{m_{vic}\} = u.TL'_k$ - $\{m_{new}\}$ the subset of k-1 results for u that do not change when m_{new} replaces m_{vic}. By developing (4) and simplifying the common part that corresponds to $u.F_k$, the update condition becomes:

$$dr_u(m_{new}, u.F_k) > dr_u(m_{vic}, u.F_k) \tag{5}$$

We note $dr_u(m, X) = \nu\ rel(m, u) + (1-\nu)\ \frac{2}{k-1} D_m(X)$ the *simplified relevance-diversity scoring function*, where $D_m(X) = \sum_{x \in X} dist(m, x)$ can be interpreted as the *diversity of the set X relative to message m*. $dr_u(m, X)$ combines the relevance of m for u with the diversity of X relative to m. Note that evaluating condition (5) is significantly faster than for the equivalent condition on DR.

Victim selection heuristics. We explore two heuristics for choosing the victim message in $u.TL_k$: (1) *Minimum relevance* (MR), which selects the message with the smallest relevance to u: $m_{vic} = argmin_{m \in u.TL_k} rel(m, u)$. (2) *Minimum relevance-diversity* (MRD), which introduces a part of diversity into the heuristics, by selecting the message with the smallest simplified relevance-diversity dr_u: $m_{vic} = argmin_{m \in u.TL_k} dr_u(m, u.TL_k - \{m\})$.

The DA-SANTA index. Figure 1 presents the DA-SANTA index structure, as an extension of the SANTA index, composed of sorted lists of users by profile term t_i (text index), by user importance for u_j (social index) and by relevance score limit μ. DA-SANTA adds an extra list η to handle diversity, as follows.

Like for SANTA, we consider a monotonic objective function F_{DA} for the threshold-based strategy, issued from the update condition (5). Here $F_{DA}(m_{new}, u) = dr_u(m_{new}, u.F_k) - dr_u(m_{vic}, u.F_k)$, so (5) is equivalent to $F_{DA}(m_{new}, u) > 0$. By developing dr_u, the update condition becomes:

$$\nu\ rel(m_{new}, u) + (1-\nu)\ \frac{2}{k-1} D_{m_{new}}(u.F_k) - \nu\ \mu_u - (1-\nu)\ \frac{2}{k-1} \eta_u > 0 \tag{6}$$

```
DA-SANTA algorithm
Input: message m_new, index I, user table U
  On m_new publication
    for all c ∈ getCandidates(I, m_new) do
      ue ← getUserEntry(U, c.user)
      if c.upperbound > ue.dr_vic then
        s ← compute-dr(ue, m_new)
        if s > ue.dr_vic then
          ue.TL_k ← ue.TL_k ∪ {m_new} − {ue.m_vic}
          ue.m_vic ← heuristics(ue.TL_k) //MR or MRD
          ue.dr_vic ← compute-dr(ue, m_vic)
          Update I.μ, I.η
        end if
      end if
    end for
```

```
getCandidates method
Input: message m_new, index I
  initTraversal(I,m_new)
  result ← ∅
  threshold ← F+_DA(m_new)
  while threshold > 0 do
    u ← nextIndexUser(I)
    result ← result ∪ {(u, dr(m_new))}
    threshold ← F+_DA(m_new)
  end while
  return result
```

Fig. 2. The DA-SANTA algorithm

Here μ_u=$rel(m_{vic}, u)$ is the relevance of the victim for u and η_u=$D_{m_{vic}}(u.F_k)$ the diversity of $u.F_k$ relative to m_{vic}. As the choice of m_{vic} is independent of m_{new}, μ_u and η_u are independent from m_{new}, can be computed in advance and maintained after each top-k update.

Like for SANTA, the term in $rel(m_{new}, u)$, when using scoring functions such as (2) or (1) with cosine similarity, is indexed by the text and social indexes. We also have the term in μ_u, indexed by the min-score index, with the difference that the indexed value is here $-rel(m_{vic}, u)$. For the term in η_u, we add a new list η to the index (diversity index), organized like μ but storing the values of $-\eta_u$ in descending order. However, the term in $D_{m_{new}}(u.F_k)$ in (6) cannot be indexed in a similar way. Therefore, we consider an upperbound of $F_{DA}(m_{new}, u)$, by replacing $D_{m_{new}}(u.F_k)$ with k-1, given that $D_{m_{new}}(u.F_k)$ sums k-1 distances $\in [0, 1]$. We note this upperbound F^+_{DA}. With relevance function (2) using cosine similarity, $score(m, u) = a \sum_{t_i \in m} w_{im} w_{iu} + b\, f(u, u^m) + c\, G(m)$, we obtain the following objective function, monotonic in the (underlined) index dimensions:

$$F^+_{DA}(m_{new}, u) = \nu\ (a \sum_{t_i \in m_{new}} w_{im_{new}} \underline{w_{iu}} + b\, \underline{f(u, u^{m_{new}})} + c\, G(m_{new})) + 2(1-\nu) - \nu\, \underline{\mu_u} - (1-\nu)\, \underline{\frac{2}{k-1}\eta_u}$$

DA-SANTA also manages a *user table* to keep for each user in the social network the current $u.TL_k$ and information for score computation.

The algorithm. Figure 2 presents DA-SANTA. On publication of a new message m_{new}, the *getCandidates* method returns only users that have a chance to integrate m_{new} in their top-k. Each returned candidate is a couple (user, upperbound) - we take advantage here of the capability of the index traversal method to also estimate an upperbound for $dr_u(m_{new}, u.F_k)$ (here u is $c.user$).

For each candidate, its entry ue in the user table is necessary to compute the real value of $dr_u(m_{new}, u.F_k)$. To avoid as much as possible this costly operation, we filter out cases when the upperbound is not greater than $dr_u(m_{vic}, u.F_k)$ (stored in $ue.dr_{vic}$). After computing the real score with the *compute-dr* function, if the update condition (5) is fulfilled, we update the top-k list $u.TL_k$, select the new victim by using heuristics MR or MRD, and update $dr_u(m_{vic}, u.F_k)$.

Finally, we update the index lists μ and η, by moving only entries for u, following the new value of $-rel(m_{vic}, u)$, respectively $-D_{m_{vic}}(u.F_k)$.

The *getCandidates* method traverses the index to prune candidates. Given m_{new}, *initTraversal* selects the related lists from the index and computes the coefficients of the objective function $F^+_{DA}(m_{new}, u)$. The index lists traversal may follow any threshold algorithm strategy (e.g. TA[6]) through the call to

nextIndexUser, which returns the next user (in some of the lists) not yet seen in the index (new candidate).

The threshold is the maximal value that the objective function F^+_{DA} may have, and is evaluated by $\overline{F^+_{DA}}(m_{new})$ as being $F^+_{DA}(m_{new}, u)$ applied to the last visited value in each index list. The monotony of F^+_{DA} and of the index lists implies that for a new candidate u, $F^+_{DA}(m_{new}, u) \leq \overline{F^+_{DA}}(m_{new})$. For the same reasons, we obtain an upperbound for $dr_u(m_{new}, u.F_k)$ through $\overline{dr}(m_{new})$, computed like $\overline{F^+_{DA}}(m_{new})$ but only on the part that corresponds to $dr_u(m_{new}, u.F_k)$. Each new candidate and its upperbound for dr_u are appended to the results list.

Index traversal stops when the decreasing threshold becomes ≤ 0.

4 Experimental evaluation

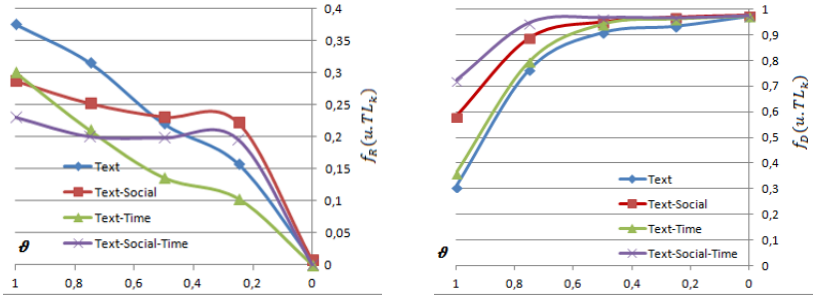
Experimental setting. Our settings are similar to those applied in [1]. The social network is extracted from Twitter, with about 104 000 users and 18 million direct links between them. Computation of f uses the existence of a direct link (u_1, u_2) and the number of actions of u_1 on the messages of u_2. We use about 500 000 tweets extracted from the last 200 tweets for each user. Messages contain 3-4 terms in average. A dictionary of about 187 000 terms was built with message terms employed by at least 5 users. For each user, the profile contains all the dictionary terms that occur in his messages - the average profile size is 125. Note that user profiles and f are not continuously recomputed.

The relevance scoring function (1)(2) uses the default coefficients a=0.5, b=0.375 and c=0.125, while $G(m)$ uses the Klout score. Time bonus uses a linear function $TB(t^m - t_o) = 1 + (t^m - t_o)/T_b$, where T_b is the period of time after which an extra bonus equal to the initial $score(m, u)$ is earned, with a default value of 15 days. We consider four combinations of factors in the relevance scoring function: *Text-Social-Time* corresponds to the complete function (1), *Text-Social* to (2), *Text-Time* ignores the social components considering b=c=0, and *Text* only keeps the text relevance.

The other default values in the experiments are k=10 and ν=0.75.

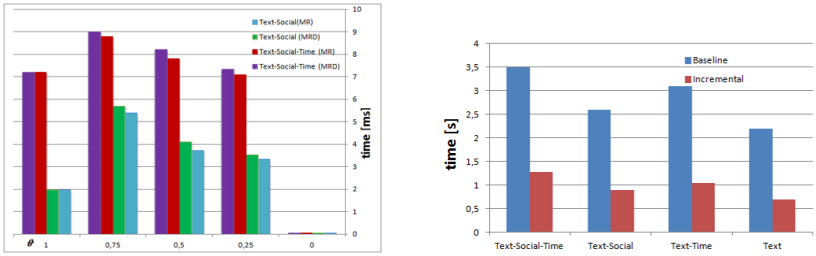
We compare DA-SANTA with two other algorithms. *Baseline* corresponds to the *basic algorithm* (Section 2), *Incremental* [9] optimizes the computation of the relevance-diversity scores by using condition (5) with dr_u instead of (4). All the algorithms have an *initialization phase* that processes the first 300 000 messages, followed by *the measure phase* on the remaining 200 000 messages.

Effectiveness. We measure the quality of the results in terms of relevance ($f_R(u.TL_k)$) and diversity ($f_D(u.TL_k)$), while varying the balance parameter ν. The values of f_R and f_D are normalized to [0,1] by division by k. Figure 3 represents this variation for MR and for each type of relevance scoring. Values for ν=1 correspond to the case without diversity, while ν=0 is the other extreme case, where only diversity counts. Figure 3.a) shows a monotonic decrease of relevance in all the cases when ν decreases, to very low values when ν=0. However, when social criteria are included into the relevance scoring, the decrease is much smoother. This can be explained by a better natural content-diversity of messages when the relevance is not only based on content. Note also that relevance scores are not comparable among the various scoring types.



(a) Variation of relevance (b) Variation of diversity

Fig. 3. Variation with ν of the achieved relevance and diversity, with MR heuristics



(a) DA-SANTA with MR and MRD (b) *Baseline* and *Incremental*

Fig. 4. Execution time for DA-SANTA, *Baseline* and *Incremental*

Figure 3.b shows that diversity grows when ν decreases, with a stabilization to high values around ν=0.6. We notice that relevance functions including more criteria provide increased content diversity. Also, the social network criteria appear to have a good influence on diversity, better than the time bonus.

When using the MRD, measures are very close to those for MR, with noticeable better diversity, although the difference with MR is small.

In conclusion, a small contribution of diversity to the balance with relevance brings a very good diversity to the results, without loosing much of the relevance.

Efficiency. We measure the execution time per message, for both MR and MRD. Since time-dependent scoring has a significant impact on the execution time (because of the increased probability of new messages to be relevant and to enter the top-k), we compare two scoring cases, without (*Text-Social*), and with (*Text-Social-Time*) time bonus.

Figure 4.a presents the variation of the execution time with ν. In all the cases, the execution time first increases when ν decreases from 1 to around 0.75, then it decreases when ν continues to decrease. The initial increase is explained by the increasing role of the diversity in the global score, provoking more and more updates to the top-k and to the index. Around ν=0.75 the diversity becomes high enough and cannot increase too much anymore; the diversity part in the objective function F^+_{DA} becomes important enough to produce a quicker termination of the index traversal. In all the cases, MR is slightly faster than the MRD. Time-

dependent scoring has a much higher impact on the execution time, which is 1.5 to 2.5 times longer with *Text-Social-Time* than with *Text-Social*.

In conclusion, the execution time of DA-SANTA (ms per message) is adapted to continuous top-k processing. Time-dependent scoring has a real impact on the execution time, but do not change the order of magnitude. The victim selection heuristics has less impact than the other efficiency factors.

Comparison with *Baseline* and *Incremental*. *Baseline* and *Incremental* produce both the same relevance and diversity, since they test each time all the possible victims in the top-k. Comparing DA-SANTA with them evaluates the loss of relevance and diversity by applying a victim selection heuristics. Measures (not shown here for space reasons) indicate a negligible loss of relevance and diversity, which proves the very good quality of results produced by DA-SANTA.

Figure 4.b compares the execution time of *Baseline* and *Incremental* with all the DA-SANTA scoring cases. *Incremental* is about 3 times faster than *Baseline*, but unlike DA-SANTA, its execution time (about 1 second/message) is not appropriate for continuous processing of top-k queries at a social network scale.

In conclusion, DA-SANTA delivers a similar quality of results 2-3 orders of magnitude faster than *Baseline* and *Incremental*, with execution times compatible with the continuous processing of top-k queries in large social networks.

References

1. A. Alkhouli, D. Vodislav, and B. Borzic. Continuous top-k queries in social networks. In *CoopIS '16*, pages 24–42, 2016.
2. L. Chen and G. Cong. Diversity-aware top-k publish/subscribe for text stream. In *SIGMOD '15*, pages 347–362, New York, NY, USA, 2015. ACM.
3. M. Drosou and E. Pitoura. Diversity over continuous data. *IEEE Data Eng. Bull.*, 32(4):49–56, 2009.
4. M. Drosou and E. Pitoura. Search result diversification. *SIGMOD Record*, 39(1):41–47, 2010.
5. M. Drosou and E. Pitoura. Dynamic diversification of continuous data. In *EDBT '12*, pages 216–227, New York, NY, USA, 2012. ACM.
6. R. Fagin. Combining fuzzy information: An overview. *SIGMOD Rec.*, 31(2):109–118, June 2002.
7. S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *WWW '09*, pages 381–390, New York, NY, USA, 2009. ACM.
8. P. Haghani, S. Michel, and K. Aberer. The gist of everything new: Personalized top-k processing over web 2.0 streams. In *CIKM '10*, pages 489–498, 2010.
9. E. Minack, W. Siberski, and W. Nejdl. Incremental diversification for very large sets: A streaming-based approach. In *SIGIR '11*, pages 585–594, 2011.
10. K. Mouratidis and H. Pang. Efficient evaluation of continuous text search queries. *IEEE Trans. on Knowl. and Data Eng.*, 23(10):1469–1482, Oct. 2011.
11. W. Rao, L. Chen, S. Chen, and S. Tarkoma. Evaluating continuous top-k queries over document streams. *World Wide Web*, 17(1):59–83, Jan. 2014.
12. A. Shraer, M. Gurevich, M. Fontoura, and V. Josifovski. Top-k publish-subscribe for social annotation of news. *Proc. VLDB Endow.*, 6(6):385–396, Apr. 2013.
13. N. Vouzoukidou, B. Amann, and V. Christophides. Processing continuous text queries featuring non-homogeneous scoring functions. In *CIKM '12*, pages 1065–1074, 2012.