Continuous Top-k Processing of Social Network Information Streams: A Vision

Abdulhafiz Alkhouli, Dan Vodislav, Boris Borzic

Continuous top-k processing of social network information streams: a vision

Abdulhafiz Alkhouli[1], Dan Vodislav[1], and Boris Borzic[1]

ETIS, ENSEA / University of Cergy-Pontoise / CNRS, Cergy, France

Abstract. With the huge popularity of social networks, publishing and consuming content through information streams is nowadays at the heart of the new Web. Top-k queries over the streams of interest allow limiting results to relevant content, while continuous processing of such queries is the most effective approach in large scale systems. Current systems fail in combining continuous top-k processing with rich scoring models including social network criteria. We present in this paper our vision on the possible features of a social network of information streams, with a rich scoring model compatible with continuous top-k processing.

Keywords: information streams, social networks, continuous top-k query processing

1 Introduction

The advent of Web 2.0 technologies has deeply changed the way information is published and consumed on the Web. Passive readers have become both active information collectors and producers, while dynamic content generation and consumption has continuously gained importance compared to traditional Web publishing (of Web pages) and exploring (through bookmarks, search engines and hyperlink navigation).

Content publishing takes more and more the form of *information streams* available through various information channels: RSS/Atom feeds from newspapers and media, blogs, discussion forums, social networks, etc. Information streams consist of flows of items, usually short semi-structured text messages, possibly containing links to some Web resources (images, videos, pages, etc.), and continuously published through specific diffusion channels. Users may subscribe to several information channels of interest and continuously receive on it, in real-time, new published content. With the huge popularity of social networks and of other information stream sources, this method of publishing and consuming content is today at the heart of the new Web.

An important dimension in this publish-subscribe (pub/sub) framework is the relationship between publishers of information streams and subscribers. This *social network dimension* varies from no relationship at all in the case of RSS/Atom feeds, to possible interaction with the published messages on blogs (comments) and discussion forums (reply messages), and to explicit relationships between users playing the double role of publishers and subscribers on social networks such as Facebook (symmetric "friendship" relations) or Twitter (asymmetric "following" relations). The social network dimension contributes not only with providing information streams of interest to end users, but also comes with criteria to measure this potential interest.

If this pub/sub approach in content dissemination has many advantages in facilitating the access to continuously delivered, fresh, pertinent information, it also raises some significant challenges. Maybe the most important one is *the huge amount of information* available on today information channels; for a regular user, the number of potentially interesting information streams combined with the flow of messages they deliver, leads to overwhelming amounts of information. Even if channels are organized by thematic criteria to help users choosing information streams of interest, not all the published content is useful or equally useful to them. The first challenge is then to define models for filtering and ranking content, and to provide easy to use subscription languages and tools for managing delivered information.

One way to organize the large amounts of stream messages is to define *a ranking model* based on *the importance of a message relative to a given subscription query*. Measuring this importance by a score allows end users to identify and to focus on the most important messages for them, e.g. those with a score over a given threshold, or the k most important ones (top-k). The ranking model may depend on various context factors, among which we emphasize the following ones:

- *Content* based factors, measuring the adequacy of the message content with the subscription query. Since textual content is characteristic to information streams, content-based subscription queries are usually based on sets of terms of interest, and the importance of a message is evaluated from an information retrieval perspective, as the relevance of the text message to the query, based on popular models such as tf-idf [26] or BM25 [17].
- *User* based factors, measuring the importance of users and of their relationships in the social network, for instance the importance of the message publisher and of the relationship between the subscriber and the publisher. In most cases user-based importance is measured on the social network graph, by evaluating e.g. node centrality and distance between nodes.
- *Interaction* based factors, measuring the importance of messages by the reaction they provoked, expressed through actions of other users on that message. Depending on the social network context, current actions may be likes, comments, forwards, tagging as favorite, etc.
- *Time* based factors, measuring the decrease of importance for a message as time goes by. Two main approaches are used to take into account this dimension: sliding time windows [13], resulting in dropping messages older than a given duration, and time decay functions [29][28], expressing a continuous decrease of importance.

Other context factors, that we do not consider here, may contribute to evaluate the importance of messages, such as geographic location or other information elements specific to the social network and to the pub/sub environment.

The second main challenge in the pub/sub approach for information streams is the design and implementation of *efficient processing models* at a very large scale (millions, up to billions of users and information streams). In the case of ranking models based on scoring functions, where subscription results are limited to the most important messages, the main difficulty comes from the need of continuously (re-)computing the score of every message relative to every subscription query and of subsequently main-

taining the lists of subscription results. The complexity of this task depends not only on the number of messages and queries, but also on the form of the scoring function.

Two main categories of processing models have been proposed in this context. *The static approach* is based on periodic snapshot queries over the set of published messages to get the list of important messages for each user. *The continuous approach* handles subscriptions as continuous queries reacting to new messages and to other events, in order to incrementally maintain the important messages. As illustrated by the related work below, if the continuous approach is more efficient, it also has more difficulties to handle complex scoring functions. To the best of our knowledge, the continuous methods proposed so far only explored simple scoring functions, most of the time based on the textual content, eventually combined with time factors. More complex scoring, including social network factors has been proposed, but only handled through a static approach.

This paper considers, in the general context introduced above, the problem of continuously computing top-k messages for each subscription query in a very large information stream pub/sub system, including complex scoring functions corresponding to a social network environment. We describe our vision of this problem in the context of state-of-the-art related work, and propose a general model of social network information streams covering many existing cases, a scoring model in this context including all the importance factors introduced above, and finally a processing architecture for continuous top-k processing in the defined context.

The rest of the paper is organized as follows: next section presents related work, Section 3 describes the social network model and the scoring function, then Section 4 presents an architecture for continuous top-k processing, before concluding.

2 Related Work

Information stream filtering Several approaches have been proposed to tackle the problem of reducing the amount of information received from streams by filtering their contents. If the first RSS/Atom feed aggregation tools (Google Reader, NetVibes[1], etc.) did not initially consider filtering, the need for controlling the volume and for personalizing the content of received information rapidly led to the introduction of various, complex filtering criteria, such as in Yahoo! Pipes[2].

Boolean filtering has been first proposed for information streams, using filters based on Boolean predicates. Most cases focus on text filtering through Boolean keyword predicates, in a pub/sub context. Such solutions [33][15] come with various index structures for fast detection of the subscription queries concerned by a stream input message, in the context of a large number of subscriptions. In [7], keyword subscriptions are considered in the context of a micro-blogging social network and three index structures are proposed; they use three dimensions (keywords/terms, publishers and followers) to enrich indexing with the structure of the social network.

The drawback of Boolean filtering is that the number of results may be in some cases too big or too small. *Information retrieval (IR) ranking models*, such as tf-idf [26]

[1] http://www.netvibes.com

[2] https://pipes.yahoo.com/pipes/

or Okapi BM25 [17] provide ranking of results through a relevance score computed for each message in the context of a given text query. Relevance scores and ranking allow selecting the best results and adapting their number to the end user needs.

Two main approaches have been proposed for filtering stream messages in an IR ranking context. The first one uses *a predefined threshold* for the relevance score [32] [34] [6] [25]. However, finding the right threshold in a given context is a difficult task and [36] proposes a method for adaptive detection of this threshold. More recent work [21] [14] [24] [29] [28] has adopted the second approach, of *top-k computation*, by considering only the k most relevant results in a continuous processing approach. The additional difficulty in this case, compared to threshold-based ranking, is to continuously maintain a changing list of top-k results.

We mention here also some work on *filtering data (non-textual) streams*, where items are composed of vectors of typed values, numerical in most cases. Boolean filtering is considered in [3] [10] [1], where various indexes for Boolean data predicates from subscription queries are proposed. Ranking for top-k filtering in this context [8] [20] is based on multidimensional indexes for numerical vectors and faces the same problems related to the curse of dimensionality as for multimedia features indexing, which limits the number of dimensions for which these structures are efficient.

In this context, our work addresses top-k filtering for information streams in a social network environment, going beyond text-only messages. The social network parameters and possibly non-textual message components require solutions which combine IR text ranking methods with specific index structures.

Score model In ranking models for information streams, the importance of a stream message for a subscription query has been generally considered in the context of text messages and queries, based on IR text relevance models such as tf-idf and BM25. To this query-dependent score model, some approaches have also added a *global, query independent importance of messages*, based on the PageRank score [22] when messages refer web pages, on information novelty [12], on source authority [9] [16] or on user attention [30].

The social network context has been considered in the scoring models, in order to improve the relevance of subscription query results by taking into account the relationships between publishers and subscribers. Social network components are included in the score model in several approaches, such as the distance in the social graph [2] [35], user actions [18] or spatial information [31].

However, the complexity of these scoring models prevented their use for continuous top-k processing. Either they are only proposed to provide a better relevance estimation in social network environments, or, at best, they come with efficient algorithms for score components computation (e.g. distance in graph) and with static, snapshot-based algorithms for top-k evaluation [31] [35]. To the best of our knowledge, the only work on continuous top-k processing for information streams including a social network component in its score model is [29], but this is limited to the simplest component, a global, query independent importance of each message.

In the context of *social tagging networks*, such as Delicious or Flickr, score models with social network components have also been proposed. [27] and [19] consider score

models combining text and social relevance, and provide snapshot-based algorithms for top-k computation.

However, we do not consider tagging networks as producing information streams; even if some analogy may be considered between documents/tags in tagging networks and messages/actions in our information stream networks, there are too many differences between their models to generalize a realistic common social network model.

In this context, our work aims at proposing a rich score model, including social network components, providing a good compromise between expressiveness and complexity for continuous top-k processing of information streams.

Continuous top-k processing The closest work to our approach concerns continuous top-k processing models for information streams. [23] is an early work on probabilistic models for continuous top-k processing with a time sliding window w (top-k/w publish-subscribe), independently on a scoring model. [21] proposes a solution for top-k/w publish-subscribe over text message streams based on classical tf-idf cosine similarity. It uses two inverted text indexes, one for the most recent messages (in the sliding window) and the second one for the subscription queries. Top-k processing is based on the Threshold Algorithm (TA) [11] exploiting the text indexes. However, since messages are indexed, a high arrival rate results here in expensive index updates.

[14] also tackles top-k/w publish-subscribe on text information streams and proposes the COL-Filter algorithm and an improved variant POL-Filter. COL-Filter only indexes subscription queries but uses a score-oriented order for the inverted lists instead of query-oriented order in [21]. More precisely, a list for a query term τ indexes queries q containing τ, ordered by the ratio between the importance of τ in q and the current k-th best score for that q. This allows efficient top-k processing by using the TA algorithm on the index lists, but suffers from a relatively high number of updates subsequent to k-th best score changes. Message exit from the time sliding window also results in updates to the top-k results.

In a similar context, [24] proposes a strategy for sharing effort among queries in the top-k computation process, based on a covering relationship between subscription queries and an associated graph index, resulting in efficient top-k processing.

[28] proposes an adaptation of two IR top-k retrieval strategies to information streams: the document-at-a-time (DAAT) algorithm WAND [4] and the term-at-a-time (TAAT) algorithm of Buckley and Lewit [5]. Instead of time sliding windows, an continuous order-preserving decay function is proposed to handle time-dependent scoring, which eliminates the problem of top-k recomputing upon message expiration.

Unlike the above approaches considering text information streams with monotonic and homogeneous scoring functions, [29] introduces a simple social network factor in scoring: a global importance of each message, that may be based on social network criteria. This results in non-homogeneous scoring functions, where methods proposed by the approaches above are not applicable. They use a two-dimensional inverted query indexing scheme and explore efficient score bounds which drastic pruning of the search space. Like for [28], time-dependent scoring is handled through decay functions.

Excepting the last approach, all these continuous top-k processing techniques are limited to simple text scoring functions. We aim at extending these techniques to scoring functions including rich social network components.

3 Data model and scoring function

As mentioned above, we consider the problem of continuously computing the k most important messages for each user in a social network, coming from information streams published by other users in the network.

We first propose a general social network model that covers many cases of popular social network environments. Then, based on this model, we propose a general scoring function for the importance of a published message relative to a given user.

3.1 Information stream social networks

Definition 1. *An information stream social network $\mathcal{S}$ is a tuple $\mathcal{S} = (U, R, p, sim, f, s)$, where:*

- *U is a set of users.*
- *$R = \{(u_1, u_2) | u_1, u_2 \in U, u_1 \neq u_2\}$ is a set of non-symmetric relations between users; $(u_1, u_2) \in R$ means that u_1 "follows" the messages published by u_2.*
- *$p : U \rightarrow \mathcal{D}$ is a function associating to each user a profile. User profiles and message contents are both modeled as descriptive documents in $\mathcal{D}$.*
- *$sim : \mathcal{D}^2 \rightarrow [0, 1]$ is a function measuring the similarity between two descriptive documents.*
- *$f : U^2 \rightarrow [0, 1]$ is a function associating to each couple of users (u_1, u_2) the importance of u_2 for u_1 in the social network.*
- *$s : U \rightarrow \mathcal{I}$ is a function associating to each user the information stream generated by that user.*

Note that U and R respectively define the nodes and edges of the directed social network graph. To represent symmetric networks such as Facebook, two edges must be created between any related nodes u_1 and u_2: (u_1, u_2) and (u_2, u_1).

The structure of descriptive documents in $\mathcal{D}$, which model both user profiles and message contents, depends on the nature of messages. Intuitively, the profile document gathers the elements of interest for the user in messages. In the common case of text messages, where similarity is evaluated through vector models like tf-idf, a descriptive document $d \in \mathcal{D}$ may be represented as a vector of terms belonging to a dictionary $\mathcal{T}$, with a tf-idf weight associated to each term, i.e. $d = \{(t, w) | t \in \mathcal{T}, w \in \mathbb{R}^+\}$.

At the same time, a user profile represents *the subscription query* for that user. For instance, users are interested in messages whose contents is relevant to their profiles.

The sim function measures the similarity between descriptive documents. For a message content mc and a user u, $sim(mc, p(u))$ measures the interest of user u (whose profile is $p(u)$) for message of content mc. For instance, the tf-idf similarity between documents d_1 and d_2 is measured by the cosine between their vectors of weights.

Note that the user relative importance function f is defined for any couple of users in the network graph, not only for those directly related through R. Like R, f is asymmetric, generally $f(u_1, u_2) \neq f(u_2, u_1)$. Depending on the context and on the design choices, the values of $f(u_1, u_2)$ may depend on many factors, such as the paths connecting u_1 to u_2 in the graph, the similarity of the two user profiles, the actions of u_1

on the messages of u_2, etc. Consequently, the values of f may vary in time, with the creation/deletion of users and relations, with profile changes, new interactions, etc.

The choice of introducing f as a global function, characterizing any couple of users, corresponds to our intention to go beyond locality in social network relationships. While in most social networks one only sees streams published by "friends" (users to which one is explicitly connected), we aim at providing users with both *a local view* (messages from the user's "community") and *a global one* (from the rest of the network).

The information streams published by users are defined as follows.

Definition 2. *An information stream $I \in \mathcal{I}$ is a couple $I = (M, A)$, where:*

- $M = \{(ts, mc) | ts \in TS, mc \in \mathcal{D}\}$ *is a set of messages, where ts is the timestamp of the message and mc is the descriptive document of the message contents.*
- $A = \{(ts, u, m, type, ac) | ts \in TS, u \in U, m \in M, type \in AT, ac \in \mathcal{D}\}$ *is a set of actions on the stream messages. ts is the action's timestamp, u the user that realized it, m the target message of the action, $type$ its type among a set of predefined action types AT, and ac the descriptive document of the action contents.*

Stream messages and actions are implicitly ordered by their timestamps. Actions are always associated to a message and may have various types. Note that an action is not a message, even if some of them (e.g. comments, retweets) may be similar to messages in contents and in the way they are produced - all social networks provide the mechanisms to make this distinction. Examples of actions in the particular case of Twitter are retweets, replies, favorite marks on tweets, etc.

3.2 Scoring function

In the context of an information stream social network $\mathcal{S}$, the ranking of messages is driven by a scoring function that expresses the importance of a message for a user. We propose a general form of the scoring function, taking into account not only content-based factors, but also social network and time factors.

Note that we consider this scoring function *in the context of continuous top-k processing of social network information streams*. As shown in Section 2, existing work in the same context only considered simple scoring functions, with practically no social network components.

We propose here a complex scoring function, including social network factors, but still adapted to continuous top-k processing. We first present a general form for the scoring function, then we give in Section 4 some hints on how such scoring functions may be handled for continuous processing.

Definition 3. Scoring function *For a user $u \in U$ and a message m published by another user $u_m \in U, u_m \neq u$, the scoring function $score : \mathcal{M} \times U \rightarrow \mathbb{R}_+$ expresses the importance of message m for user u and has the following general form:*

$$score(m, u) = \mathcal{F}_g(CS(m, u), SS(m, u)) \quad (1)$$

- *$CS(m, u)$ expresses the content similarity between the contents of m and the profile of u. In our information stream social network model, $CS(m, u) = sim(m.mc, p(u))$.*

- *$SS(m, u)$ expresses the importance of message m to user u in the social network context.*
- *$\mathcal{F}_g$ is a monotonic aggregation function combining the content-based and the social network scores.*

The social network scoring component may take into account both *user* related and *interaction* related factors in the social network.

$$SS(m, u) = \mathcal{F}_s(US(m, u), AS(m, u)) \tag{2}$$

The monotonic aggregation function $\mathcal{F}_s$ combines the partial scores given by the user-related factors ($US(m, u)$) and by the interaction-related factors ($AS(m, u)$).

The user-related scoring function $US(m, u)$ may itself take into account two kind of factors, related to the message publisher or to the relation between the publisher and the potential receiver.

$$US(m, u) = \mathcal{F}_u(UI(u_m), UR(u, u_m)) \tag{3}$$

Here, the $UI(u_m)$ component expresses the global importance of the message publisher u_m in the social network, while $UR(u, u_m)$ measures the importance of u_m for u. They are combined through the $\mathcal{F}_u$ monotonic aggregation function. Since in our information stream social network model the relative importance of users is measured by the f function, we may consider that $UR(u, u_m) = f(u, u_m)$. UI may be based e.g. on measures of influence in the social network, such as the Klout score.

Similarly, the interaction-related scoring function $AS(m, u)$ has a global part related to message m and a part giving the importance of the interactions with m from the perspective of user u.

$$AS(m, u) = \mathcal{F}_a(AI(m), AR(m, u)) \tag{4}$$

Here, $AI(m)$ expresses the importance of message m coming from the interaction it provoked globally in the network. $AR(m, u)$ measures the importance of the interactions with the message from the perspective of user u. Intuitively, an action on m is important for u if it is done by a user u_a important for u. AI and AR, combined through the $\mathcal{F}_a$ monotonic aggregation function, may be modeled by various functions, increasing with the number of actions on the message.

If we consider the common case of linear aggregation functions, expressed as positive weighted sums, Formulas 1-4 result into the following form for the scoring function, which gives a better overview of its various components:

$$score(m, u) = \alpha CS(m, u) + \beta_1 UI(u_m) + \beta_2 f(u, u_m) + \gamma_1 AI(m) + \gamma_2 AR(m, u) \tag{5}$$

Note that state-of-the-art proposals for continuous top-k processing only consider the CS component, excepting [29] which also includes the AI and UI components.

Definition 4. Time dependent scoring function *For a message m published at time t_m, the variation in time of the importance of message m for user u is expressed by*

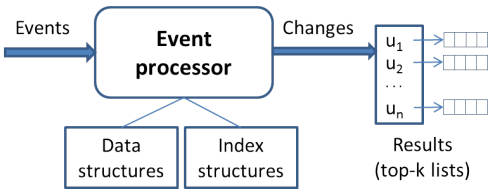

Fig. 1. General architecture for continuous top-k processing of information streams

the time-dependent scoring function $tscore : \mathcal{M} \times U \times TS \rightarrow \mathbb{R}_+$ *such that for any moment* $t \geq t_m$*:*

$$tscore(m, u, t) = score(m, u) \cdot TD(t - t_m) \tag{6}$$

- $score(m, u)$ *is the scoring function from Definition 3 and expresses the initial importance of message* m *for user* u *at moment* t_m*.*
- $TD : \mathbb{R}_+ \rightarrow [0, 1]$ *is a decreasing function such that* $TD(0) = 1$*.* TD *expresses the decrease in time of the importance of message* m*, by associating to each time duration since the message publishing, a decrease factor in* $[0, 1]$*. For instance* $tscore(m, u, t_m) = score(m, u)$ *and if* $t_1 > t_2 \geq t_m$ *then* $tscore(m, u, t_1) \leq tscore(m, u, t_2)$*.*

We make here the common choice of a message and user independent time function, which greatly facilitates message query processing, as illustrated in the next section.

Moreover, we only consider *order-preserving decay functions* for TD, i.e. functions which guarantee that the relative order of message scores is preserved in time. More precisely, an order preserving decay function TD guarantees that if at some moment t we have $tscore(m_1, u_1, t) \leq tscore(m_2, u_2, t)$, then $tscore(m_1, u_1, t') \leq tscore(m_2, u_2, t'), \forall t' > t$. Order-preserving decay functions facilitate continuous top-k processing by preserving in time the relative order of messages in the top-k lists.

In the particular case of a time dependent factor handled by *a sliding time window* of size w_t, $TD(d) = 1$ if $d \leq w_t$ and $TD(d) = 0$ when $d > w_t$. Note than in this case TD is not order-preserving, a message exiting the sliding time window becoming less important than any message still in the window.

4 Processing model

The general scoring function for social network information streams given by Definition 3 and Formulas 1 to 4, or by its linear expression 5 provides a rich model for importance evaluation compared to the state-of-the-art methods. The general function may be instantiated in many ways, with an impact on the processing method. The description of a complete solution in a specific case is out of the scope of this paper, but we present here a general approach for continuous top-k processing with such a scoring model.

Figure 1 presents the general architecture of continuous top-k processing of information streams, behaving as an event-based system. The result of such a process is the

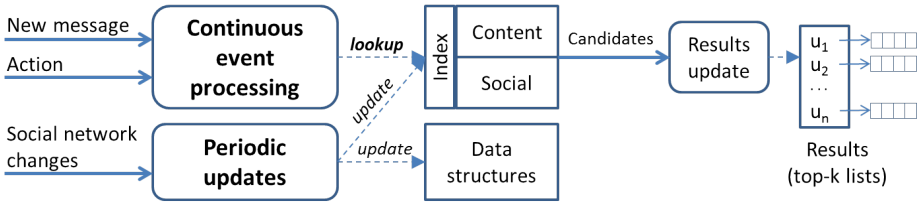

Fig. 2. Architecture for continuous top-k processing of social network information streams

set of top-k messages for each user in the social network, continuously maintained by the system. The event processor handles every input event that may produce changes to the result lists, computes changes and subsequently updates the result lists. Change computation is based on the data structures representing the information streams and the social network, and on the index structures that enable efficient event processing.

In our information stream context, we distinguish two categories of events:

- *Continuously handled events*, with potentially strong impact on top-k update, and that must be processed on the spot. We include in this category, the publication of a new message and the interaction with an existing message.
- *Secondary events*, with a weaker impact on the top-k lists; they may be accumulated and processed from time to time. We include in this category changes in the social network that may produce small changes in the scoring parameters.

Evaluating the impact of various categories of events depending on the scoring model is a difficult problem, but continuously reacting to any event that may change some message score component is not realistic in practice, given the complexity of our scoring function. The above classification of events is a necessary trade-off between efficiency and precision.

Figure 2 presents the proposed architecture for continuous top-k processing of social network information streams with a scoring function such as (1). New message publishing and actions on messages are the only continuously handled events. They provoke a lookup in the index structures, composed of a content-based index and a social index. The result of this lookup is a subset of candidates for the top-k update. The role of the index is to drop from this candidate list as many users not impacted by the event as possible, in order to enable efficient top-k processing.

Consider the linear form of the scoring function given by Formula 5 and let us note $\mu(u)$ the k-th score in the result list of user u. A new published message m has null score components for AI and AR (no action yet on m), so the only users u that must update their top-k list are those with $score(m,u) > \mu(u)$, i.e. with $e(u) = \alpha CS(m,u) + \beta_1 UI(u_m) + \beta_2 f(u,u_m) - \mu(u) > 0$. The design of the index structure must be based on the form of the $e(u)$ function.

In the case of a new action on message m, only the AI and AR components of the score change, i.e. $\Delta score(m,u) = \Delta AI(m) + \Delta AR(m,u)$. If for some user u, this score increase is enough go beyond $\mu(u)$ then m will enter the top-k list of u. The index structure must enable quick detection of users impacted by this score increase.

Secondary events are identified as social network modifications (new user relation, new user, profile update, etc.) that may modify the parameters of the social graph and implicitly of the scoring function, especially the relative importance function f. We consider that a periodic recomputation of the social network parameters is scheduled by the system. This operation will produce an update of the data and index structures.

The time-dependent scoring function (6) implies a continuous decrease in time of message scores, that cannot be handled in continuous top-k processing. We adopt the approach in [29] and [28] based on a strictly positive order-preserving decay functions. Instead of decreasing scores for old messages, order-preserving allows increasing scores for new messages, without changing the relative order of scores. By fixing an initial moment t_0 in the system, any new message m published at time t_m will have the score multiplied by a "time bonus" of $1/TD(t_m - t_0)$, which grows with t_m. With this approach scores do not vary with time anymore, which is compatible with continuous top-k processing.

5 Conclusion

This paper presented our vision on continuous top-k processing over information streams in a social network context. We proposed a general model of information streams social networks with a rich scoring function mixing content-based, user-based, interaction-based and time-based components. A general approach for continuous top-k processing in this context completes our contribution.

References

1. M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, and T. D. Chandra. Matching events in a content-based subscription system. In PODC '99, pages 53–61, 1999.
2. B. Bahmani and A. Goel. Partitioned multi-indexing: Bringing order to social search. In WWW '12, pages 399–408, 2012.
3. S. Bianchi, P. Felber, and M. Gradinariu Potop-Butucaru. Stabilizing distributed r-trees for peer-to-peer content routing. IEEE Trans. Parallel Distrib. Syst., 21(8):1175–1187, Aug. 2010.
4. A. Z. Broder, D. Carmel, M. Herscovici, A. Soffer, and J. Zien. Efficient query evaluation using a two-level retrieval process. In CIKM '03, pages 426–434, 2003.
5. C. Buckley and A. F. Lewit. Optimization of inverted vector searches. In SIGIR '85, pages 97–110, 1985.
6. J. Callan. Document filtering with inference networks. In SIGIR '96, pages 262–269, 1996.
7. R. Dahimene, C. Du Mouza, and M. Scholl. Efficient filtering in micro-blogging systems: We won't get flooded again. In SSDBM'12, pages 168–176, 2012.
8. G. Das, D. Gunopulos, N. Koudas, and N. Sarkas. Ad-hoc top-k query answering for data streams. In VLDB '07, pages 183–194, 2007.
9. G. M. Del Corso, A. Gullí, and F. Romani. Ranking a stream of news. In WWW '05, pages 97–106, 2005.
10. F. Fabret, H. A. Jacobsen, F. Llirbat, J. Pereira, K. A. Ross, and D. Shasha. Filtering algorithms and implementation for very fast publish/subscribe systems. SIGMOD Rec., 30(2):115–126, May 2001.
11. R. Fagin. Combining fuzzy information: An overview. SIGMOD Rec., 31(2):109–118, June 2002.

12. E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: Providing personalized newsfeeds via analysis of information novelty. In WWW '04, pages 482–490, 2004.
13. L. Golab and M. T. Özsu. Issues in Data Stream Management. SIGMOD Record, 32(2):5–14, 2003.
14. 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.
15. Z. Hmedeh, H. Kourdounakis, V. Christophides, C. du Mouza, M. Scholl, and N. Travers. Subscription indexes for web syndication systems. In EDBT '12, pages 312–323, 2012.
16. Y. Hu, M. Li, Z. Li, and W.-y. Ma. Discovering authoritative news sources and top news stories. In AIRS'06, pages 230–243, 2006.
17. K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: Development and comparative experiments. Inf. Process. Manage., 36(6):779–808, Nov. 2000.
18. A. Khodaei and C. Shahabi. Social-textual search and ranking. In Intl Workshop on Crowdsourcing Web Search, Lyon, France, April 17, 2012, pages 3–8, 2012.
19. S. Maniu and B. Cautis. Efficient top-k retrieval in online social tagging networks. CoRR, abs/1104.1605, 2011.
20. K. Mouratidis, S. Bakiras, and D. Papadias. Continuous monitoring of top-k queries over sliding windows. In SIGMOD '06, pages 635–646, 2006.
21. 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.
22. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1999.
23. K. Pripužić, I. P. Žarko, and K. Aberer. Top-k/w publish/subscribe: Finding k most relevant publications in sliding time window w. In DEBS '08, pages 127–138, 2008.
24. 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.
25. W. Rao, A. W.-C. Fu, L. Chen, and H. Chen. Stairs: Towards efficient full-text filtering and dissemination in a dht environment. In ICDE '09, pages 198–209, 2009.
26. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. Inf. Process. Manage., 24(5):513–523, Aug. 1988.
27. R. Schenkel, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. X. Parreira, and G. Weikum. Efficient top-k querying over social-tagging networks. In SIGIR '08, pages 523–530, 2008.
28. 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.
29. N. Vouzoukidou, B. Amann, and V. Christophides. Processing continuous text queries featuring non-homogeneous scoring functions. In CIKM '12, pages 1065–1074, 2012.
30. C. Wang, M. Zhang, L. Ru, and S. Ma. Automatic online news topic ranking using media focus and user attention based on aging theory. In CIKM '08, pages 1033–1042, 2008.
31. D. Wu, Y. Li, B. Choi, and J. Xu. Social-aware top-k spatial keyword search. In MDM '14, pages 235–244, 2014.
32. T. W. Yan and H. Garcia-Molina. Index structures for information filtering under the vector space model. In ICDE'94, pages 337–347, 1994.
33. T. W. Yan and H. García-Molina. Index structures for selective dissemination of information under the boolean model. ACM Trans. on Database Syst. (TODS), 19(2):332–364, 1994.
34. T. W. Yan and H. Garcia-Molina. The sift information dissemination system. ACM Trans. Database Syst., 24(4):529–565, Dec. 1999.
35. P. Yin, W.-C. Lee, and K. C. K. Lee. On top-k social web search. In CIKM, pages 1313–1316. ACM, 2010.
36. Y. Zhang and J. Callan. Maximum likelihood estimation for filtering thresholds. In SIGIR '01, pages 294–302, 2001.