EnBlogue – Emergent Topic Detection in Web 2.0 Streams

Foteini Alvanaki ♣ , Sebastian Michel ♣ , Krithi Ramamritham ◇ , Gerhard Weikum ♠

♣ Saarland University, Germany ◇ IIT Bombay, India ♠ Max-Planck Institute Informatics, Germany

♣ {alvanaki|smichel}@mmci.uni-saarland.de ◇ krithi@iitb.ac.in
♠ weikum@mpi-inf.mpg.de

ABSTRACT

Emergent topics are newly arising themes in news, blogs, or tweets, often implied by interesting and unexpected correlations of tags or entities. We present the enBlogue system for emergent topic detection. The name *enBlogue* reflects the analogy with emerging trends in fashion often referred to as *en Vogue*. EnBlogue continuously monitors Web 2.0 streams and keeps track of sudden changes in tag correlations which can be adjusted using personalization to reflect particular user interests. We demonstrate enBlogue with several real-time monitoring scenarios as well as with time lapse on archived data.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval —*information filtering, selection process*

General Terms

Algorithms, Performance

Keywords

Web 2.0 streams, emergent topics

1. INTRODUCTION

Web 2.0 streams, like blog postings, micro-blogging tweets, or RSS feeds from online communities, offer a wealth of latest news about real-world events and topics dominating societal discussions.

For example, natural disasters, military incidents, celebrity scandals, or upcoming movie premiers are sometimes covered more candidly and promptly in these forums than on official news channels. On the other hand, this wealth may easily overwhelm users. To cope with the resulting information overload, Web pages, blog postings, or tweets are often organized into fine-grained taxonomies, and users are guided by topics for search and exploration (see, e.g., [8, 2]).

Users are particularly interested in *emergent topics* that arise from recent events but which cannot be easily retrieved using existing categories. For example, someone may want to know or be alerted about "implications of the eruption of Eyjafjallajökull on the Icelandic air traffic". Immediately after the event, no category for this topic would exist in the taxonomy; it may take the Web 2.0 platform providers a few days to create categories like "Iceland air traffic" or "volcano air traffic". Of course, users can search by keywords, but the point is that they want to be automatically notified about a newly arising topic that is about to become hot - and they want to know this as soon as possible. Note that spotting such trends is very different from identifying popular topics; we are interested in sudden changes in the correlation between tags associated with the data.

The fact that these trends consist of pairs or, in general, sets of tags offers the possibility of a full exploration of social media given the detected tag set as input, for instance, in the form of a traditional keyword query. Hence, we see enBlogue as a *portal to stay tuned* on recent issues while inspecting only the novel aspects from the massive amount of available information.

EnBlogue consists also of a *personalization* component that allows users to register continuous keyword queries or to choose pre-selected topic categories to influence the nature of the emergent topics presented.

EnBlogue is online at `http://blogue.mmci.uni-saarland.de/`. To avoid overloading or attacking the web server, we have currently limited the functionality to showing emergent topics from a replayed news archive. The demo at SIGMOD shows a wider suite of application scenarios, with news, blogs, and live tweets.

We will now briefly discuss related work and then present an overview on the techniques used in enBlogue, followed by a presentation of implementation issues. Last, we describe the actual demonstration.

2. RELATED WORK

There is huge interest in analyzing Web 2.0 communities and social media. Hot topics include click-log and user-activity analysis, understanding the evolution of social interactions, opinion mining and collaborative recommendation, and the dynamics of information and influence propagation (especially in the blogosphere).

A classification based approach to detect events and the corresponding documents associated with these events has been recently addressed by Becker et al. [1]. Mathioudakis et al. [5] deal with the problem of identifying items that attract attention, by exploiting user interactions (such as

SIGMOD'11, June12–16, 2011, Athens, Greece.
ACM 978-1-4503-0661-4/11/06.

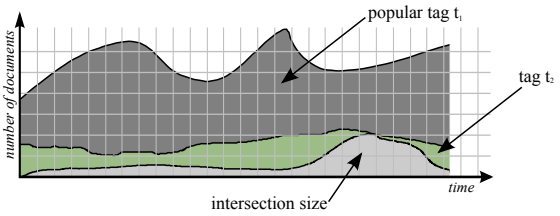

Figure 1: Interesting shift in correlation of two tags.

comments, quotes, or clicks) in social media. Potentially, boosting the influence of annotated documents identified by [5] can complement our emergent topic detection solution.

The work of Mathioudakis and Koudas [4] is closest to the enBlogue system. Their *Twitter Monitor* system, discovers topic trends in tweets, by detecting bursts of tags or tag groups. Tag groups are formed by clustering co-occurring tags or using spectral analysis. This approach is quite different from our setting: unlike looking solely for bursty tags, we detect shifts in tag *correlations* as they dynamically arise. The difference is illustrated in Figure 1, discussed in Section 3.

There is a vast amount of literature on general time series problems that are related to our problems, too. For instance, the recent work by Mueen and Keogh [6] on discovering repeating patterns in time series or work on bursts and periodic events detection over time series. Dealing with time series in this general sense is a sub-problem of our approach that arises in the second step of our framework, hence, enables us to benefit from existing solutions to this end.

The recent work by Pu et al. [7] aims at efficient entity tagging over text streams which is orthogonal to our approach, and can be easily deployed as a preprocessing operator to feed entities to the enBlogue engine thereby making it more efficient.

The Taglines project [3] considers mining the evolution of tag clouds. However, the emphasis here is on visualization, not on detecting interesting correlation trends.

Overall, despite work on related sub-aspects, our problem of identifying shifts in the correlations between taxonomic tags, as a function of time, is unique and has not been addressed by prior work.

3. APPROACH

Figure 1 shows an illustration of two tags and the corresponding overlap in terms of the number of documents (within a time window of interest) that contain both tags. The figure shows the behavior over time for two tags, a popular tag t_1 and a less popular tag t_2. By inspecting the size of the intersection over time, we see that the peaks in the popular tag have no influence on the size of the overlap. In contrast, we see that the size of the intersection grows dramatically, an explanation for which can not be given solely by looking at the individual frequencies of t_1 and t_2. Note that detecting interesting shifts is different from analyzing hot tags by themselves or mining correlation between time series representing single tags.

Our framework consists of three stages:

(i) **Seed tag selection:** As the name indicates, seed tags are used to trigger the computation in the following steps. Seed tags can be determined based on different criteria, such as popularity and volatility. We choose seed tags to be popular tags. Popularity is easy to measure as it merely requires computing a sliding-window average on the document stream. We use seed tags to generate candidate topics, i.e., pairs of tags that contain at least one seed tag. The rationale is that for a tag pair to become interesting and form an emergent topic, at least one of the two tags should be "hot" by itself. We then analyze only the correlations of the candidate topics.

(ii) **Correlation tracking:** For each tag pair that contains at least one seed tag, we keep track of their correlations. For each such pair, we continuously monitor the amount of documents that are annotated with both tags. There are multiple ways how to calculate a correlation measure that reflects some notion of interestingness. In the more complex case of documents being represented by their entire tag sets or term distributions, we can apply information-theory measures like relative entropy to assess the similarity of tag/term usage.

(iii) **Shift detection:** We consider sudden (but significant) increases in the correlation of tag pairs as an indicator for an emergent topic. We refer to such increases as shifts. We say that a shift is sudden if it cannot be predicted using the previous correlation values. Hence, at any point in time we use the previous correlation values and try to predict the current ones. If a predicted value is far away from the real one then the topic is considered to be emergent and the prediction error is used as a ranking criterion. At any point in time the score of a topic is the maximum of the current prediction error and the prediction errors from the past, dampened appropriately using an exponential decline factor with a half life of approximately 2 days. The topics are then ranked according to these scores. The topics that have bigger scores are considered more emergent and ranked higher in the final result.

The third stage provides for each considered pair a quantitative measure for the shift within a configurable time period. These values are used to rank tag pairs and to report the top-k most interesting ones, thus presenting the user with emergent topics.

Entity Tagging

To enrich incoming documents, we use an automatic entity-tagging method to extend the tag space with named entities like people, organizations, or places. These entity tags can either be handled independently of the regular tags, or alternatively combined with regular tags to detect tag/entity mixtures as emergent topics.

Our entity tagging works as follows. When a document arrives, we scan its text content with a sliding window of up to 4 successive terms, and check whether substrings of these match the title of a Wikipedia article. These checks also consider Wikipedia redirects which we use to map different namings of a single entity to one unique name. In addition, we have implemented a second filter consisting of lookups in an ontology (e.g., YAGO), which allows us to focus on particular entity types.

4. IMPLEMENTATION

4.1 Core Engine

The implementation is done in Java 1.6 and follows the standard concepts of a push-based architecture for stream processing. At the data source level, it consists of several wrappers that either consume live streams or replay existing datasets for experiments. Data is represented in form of a tuple consisting of `(timestamp, docId, set of tags, set of entities)` consumed by stream operators and pushed along producer-consumer edges in query-processing plans. The filtered and manipulated data items finally arrive at sinks in the operator DAG. One of the sinks is the operator that computes the final rankings of emergent topics and sends them to our Web server for visualization.

The enBlogue architecture is very flexible and can be easily extended. There are plug-in options for sketching operators that map stream items into synopses, statistics operators, shift prediction operators, etc. The system allows executing multiple query plans in parallel, where overlapping parts, like data sources, sketching operators, entity tagging, and statistics operators are shared for efficiency. It hence allows us to compare emergent topic rankings obtained from different parameter settings in real-time.

4.2 Front-End User Interface

The enBlogue system also has a Web-based user interface that provides real-time monitoring and user notifications in a push-based manner (i.e., without the user having to continuously poll the server for updates on emergent topic rankings). This has been implemented using AJAX technology, more specifically, the push-based variant offered by the open-source *Ajax Push Engine (APE)* (see `www.ape-project.org`). APE includes a Javascript framework for real-time data streaming to Web browsers, without any installations on the client side. Topic rankings generated by our back-end server are sent to an installation of APE which dispatches the messages to the registered clients, i.e., all Web browsers that have currently active sessions to our Web site. Due to the lightweight implementation we in particular also support (mobile) smartphone users receiving continuous updates over low bandwidth connections.

5. DEMONSTRATION DESCRIPTION

Users interact with enBlogue through its Web-based user interface. The features underlying enBlogue are demonstrated through the following three show cases:

Show case 1: Revisiting Historic Events

This show case aims at providing insights on the way enBlogue works, for known historic events. For this, we have obtained the New York Times archive, consisting of news articles from 1987 and 2007, a total of 1.8 million full-text documents. Each article is manually assigned (by the New York Times back-office) to one or more categories and annotated with additional descriptors. We use these categories and descriptors as tags.

We will pre-select historic events from certain points in time and different topic categories, such as US election issues, hurricanes, or sport events. The users can see how enBlogue ranks the topics in these categories depending on the time period we are focusing. Since all these events belong to the past, each user, according to his knowledge, experience, and interests, can judge whether the rankings would be satisfactory or not. In addition, users can specify their own time ranges and see how the ranking changes with different time periods.

Show case 2: Live Data

EnBlogue includes a set of wrappers to consume data from Twitter and several RSS feeds from blogs and online newspapers. Users can see how enBlogue ranks topics that are currently evolving. They can also watch how the rankings for these topics changes with time. Since a change in the rankings may need a significant amount of time in order to depict accurate results, we offer a time lapse view over a sliding window of the past couple of days. Patient users are welcome to wait for the live data to cause the change.

Users can also participate in an attempt to influence the rankings produced by enBlogue by creating a new topic. With the proper system configuration and the help of the present twitter users we may be able to see a topic regarding SIGMOD and Athens in a highly ranked position in the list of the emergent topics identified by enBlogue!

Show case 3: Personalization

For the above show cases we will demonstrate the usefulness of personalization. Users can provide term based descriptions of their field of interest or choose between several pre-defined topic categories. The topics will be ranked according to the specified user preferences and each user will be presented with a list containing completely different or just differently ordered emergent topics. Users can change their preferences at any time and observe the impact of the new preferences in the list with the emergent topics.

6. REFERENCES

[1] Hila Becker, Mor Naaman, and Luis Gravano. Learning similarity metrics for event identification in social media. *WSDM*, 2010.

[2] Manish Bhide, Venkatesan T. Chakaravarthy, Krithi Ramamritham, and Prasan Roy. Keyword search over dynamic categorized information. *ICDE*, 2009.

[3] Micah Dubinko, Ravi Kumar, Joseph Magnani, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Visualizing tags over time. *TWEB*, 1(2), 2007.

[4] Michael Mathioudakis and Nick Koudas. Twittermonitor: trend detection over the twitter stream. *SIGMOD*, 2010.

[5] Michael Mathioudakis, Nick Koudas, and Peter Marbach. Early online identification of attention gathering items in social media. *WSDM*, 2010.

[6] Abdullah Mueen and Eamonn J. Keogh. Online discovery and maintenance of time series motifs. *KDD*, 2010.

[7] Ken Pu, Richard Drake, Oktie Hassanzadeh, and Renee Miller. Online annotation of text streams with structured entities. *CIKM*, 2010.

[8] Anand Rajaraman. Kosmix: Exploring the deep web using taxonomies and categorization. *PVLDB*, 2(2), 2009.