Design & Implementation of Advanced Clustering Algorithm for News Feeds: RSS Aggregator

Anjani Pandey
VITS, Satna

Vinod Singh
VITS, Satna

ABSTRACT

With the development of Internet, the information which appears by text form is more and more frequent. It becomes one kind of the most easily to gain and the richest interactive resources. In recent years, different commercial Weblog subscribing systems have been proposed to return stories from users' subscribed feeds. An RSS feed may have several different topics. A user may only be interested in a subset of these topics. In addition there could be many different stories from multiple RSS feeds, which discuss similar topic from different perspectives. A user may be interested in this topic but do not know how to collect all feeds related to this topic. In contrast to many previous works, we will cluster all stories in RSS feeds into such a structure to better serve the readers. Through this way, users can easily find all their interested stories. In this paper, we propose a novel clustering-based RSS aggregator for Weblog reading from Internet.

General Terms

Clustering of News Blogs, Web Mining, RSS Aggregator

Keywords

Web Blogs, RSS Aggregator, Clustering News Blogs, Web Mining, XML

1. INTRODUCTION

RSS (Really Simple Syndication) is an Extensible Markup Language (XML) file which is used by sites for syndication of their articles on the Internet. Programs known as RSS Reader or RSS Aggregator are used for both managing the subscription to feeds and downloading articles in the subscribed feeds [5]. Obviously, not all the received articles are of interest to the users, thus a mechanism is needed to filter the irrelevant ones out and to pinpoint only those that match the user's interest. Also, there should also be a mechanism to recommend relevant articles from RSS feeds to which the user is not subscribed.

2. PROBLEM DOMAIN

An Ordinary RSS Aggregator's job is to fetch the latest news and articles and present them to users. This prevail over the users by loading large amounts of information, most of which might not be of their interest. People will be deluged by the articles which the RSS Aggregators return to them and would not be able to find the ones they are looking for easily. Several approaches have been developed to alleviate the information overload problem. These approaches are based on Adaptive Information Retrieval and artificial intelligence methodologies. A number of these approaches will be reviewed along with their strengths and weaknesses. One approach is static profiling. In this approach, users give some information about their static and predictable characteristics through the registration or survey forms [9].

Users general interests could be understood by analyzing the information they give away during static profiling. The advantage of this approach is that the system will get an idea about the users interest at the starting point. On the other hand, this approach has two problems: firstly, the information is static and will not reflect the change of the users interest in time. Secondly, this is a solely profiling method. Because the process is subjective, the interest of that user can only be predicted. Moreover, the information about a user's interests is not being used for other users with similar interest [9].

Another approach is dynamic profiling (or behavioral profiling). In this approach the behavior of the users is analyzed. Although this method is dynamic and it changes according to the current needs of the users, it does not capture the general interest of the users [9]. The third approach is content based filtering. In this approach the content of items associated with a user profile is compared to other user profiles and documents with similar content are selected. The problem in this method is that some users do not give explicit feedback about themselves. Giving feedback explicitly requires that users examine and rate an item. For many users this is a costly action. Thus, implicit rating is needed to overcome this cost.

Another approach is collaborative filtering. This method organizes people with similar interest into peer groups. A document can be recommended to all the people in the same group if it was of interest of the peers. This approach uses clustering techniques on user profiles. The effectiveness of this method is tightly related to the clustering methodology [9]. In addition, as RSS became more popular, many RSS readers and aggregators have been developed. Most of these are client-based, i.e. an application program that is downloaded and run on user's machine. These programs were developed to help users manage subscription to RSS feeds, get the latest articles for those feeds, and store them if needed. There are several issues involved in the client-based approach. Firstly, the important and favorite articles will be stored in the users' machine and can only be accessed from there. In other words, they are not accessible from anywhere except the very machine they were saved on.

3. MOTIVATION

RSS (most commonly expanded as Really Simple Syndication) is a family of web feed formats used to publish frequently updated works—such as blog entries, news headlines, audio, and video—in a standardized format.[2] An RSS document (which is called a "feed", "web feed",[3] or "channel") includes full or summarized text, plus metadata such as publishing dates and authorship. Web feeds benefit publishers by letting them syndicate content automatically. They benefit readers who want to subscribe to timely updates from favored websites or to aggregate feeds from many sites into one place. RSS feeds can be read using software called an "RSS reader", "feed reader", or "aggregator", which can be web-based, desktop-based, or mobile-device-based. A standardized XML file format allows the information to be published once and viewed by many different programs. The user subscribes to a feed by entering into the reader the feed's URI or by clicking a feed icon in a web browser that initiates the subscription process. The RSS reader checks the user's subscribed feeds regularly for new work, downloads any updates that it finds, and provides a user interface to monitor and read the feeds. RSS allows users to avoid manually inspecting all of the websites they are interested in, and instead subscribe to websites such that all new content is pushed onto their browsers when it becomes available. RSS formats are specified using XML, a generic specification for the creation of data formats.

They need to quickly scan and select interesting topics in the title and avoid duplicate articles from different publishers as they have already consumed this information. The amount of content has also been augmented, over the past few years, with the increased popularity of user generated content such as weblogs / blogs (e.g. TypePad), micro-blogs (Twitter), photo sharing services (e.g. Picasa) and socialnetworking (e.g. Facebook), as well as prevalence of web feeds from various websites. For example blog search engine Technorati reports about 1 million posts a day and more than 200 million blogs today, and LiveJournal reports about 20 millions accounts. This user-generated content is rapidly expanding as authors can touch a large number of readers with at low cost, and with little publication and marketing effort. In addition, another trend is the growing number of users that are attracted by content, rather than regularly returning to visit particular websites, i.e. they navigate directly to the article of interest either via a search engine or a web feed reader.

Web Feeds permit the subscription to regular or frequently updated content. Web content can be pushed to the subscribers, where they can subscribe to any type of web feeds of interest to them and can unsubscribe at any time. For example a user would not typically navigate daily to a blog, but would instead subscribe to be notified by email of any additions by RSS. Web feeds are not limited to news content, they include anything that is periodically updated or with a notification such as podcasts, blogs, social networking, social photo-sharing service, or publication updates. Web feeds allows users to subscribe to syndicated headline or headline-and-short-summary content which can be delivered to different channels including browsers, web portals, news readers, email, mash-ups, widgets / gadgets and mobile devices. A link to the full content as well as other metadata is also generally provided in the feed. This format has the advantage to also be machine readable allowing it to be for example embedded in other websites or collected by feed aggregators.

4. ADAPTIVE INFORMATION RETRIEVAL METHOD

Non-adaptive Information Retrieval systems return the same result for every user when they submit the same query. As in the figure 1, users have to adapt themselves with the system and document collection to find what they want. This adaptation usually happens by learning from the result and improving the query string. In an ideal Adaptive Retrieval system, the system should present what the user is interested in, without user's adaptation. This process is illustrated in figure 2. However, the real world is far from an ideal world. The more realistic scenario would be where both users and systems learn from each other to retrieve relevant information. This adaptation is based on their interaction. This scenario is illustrated in figure 3. The new Adaptive Information Retrieval methods try to decrease the required user interaction by applying more elaborate algorithms and more implicit feedback methods.

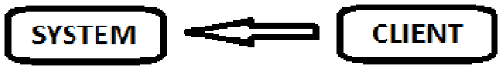

Figure 1: Users adapts themselves with the system as they interact in time. System does not adapt itself.

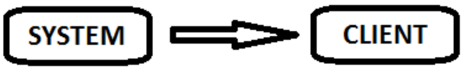

Figure 2: System becomes more adapted with users as they interact in time. Users do not adapt themselves.

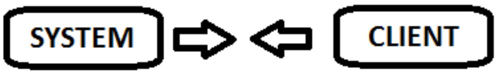

Figure 3: System and Users become more adapted with each other in time. Two-way interactions are required.

4.1 Multi-Modal (MM) approach

In [2] the authors developed a Multi-Modal (MM) approach in which the construction and the maintenance of user profiles are done automatically based on user feedback. The structure of the user profile is a set of vectors. The number, size and elements of the vectors in the profile can be changed adaptively. Two documents are considered similar if the similarity between them is less than a certain value, δ. To calculate the similarity, the cosine similarity formula will be used. If the similarity between document and profile vector is greater than δ they will be merged by the formula given below and consequently the whole profile vector will be repositioned. If similarity value is less than δ then a new profile vector will be created. On the other hand, if a profile vector is not useful anymore it will be deleted from the user profile. A vector will be considered as \not useful" if the interest represented by that vector is no longer an interest for the user.

4.2 Static Content Profiling

Static content profiling is the gathering of static information during user registration. The user profile will be created based on the information collected in the registration step. Each profile consists of a set of feature-value pairs (fi,vi) where fi is the feature keyword and vi is the value assigned to that feature.

4.3 Dynamic Content Profiling

The second approach is dynamic content profiling. In this approach the system should gather information based on the dynamic changes on the user's behavior. There are three ways to capture this information:

4.3.1. by monitoring user actions such as browsing and clicking patterns. These activities allow the interface to provide another source of information about the users.

4.3.2. by monitoring users search keywords. System adds keywords to the query implicitly to improve users' query string based on the previous similar queries that they satisfied with. These extra added keywords are usually forgotten in the next query.

4.3.3. by monitoring user preferences they built a list of interests and disinterested keywords.

4.4 Static Collaborative Profiling

Static collaborative Profiling is the third approach which helps to reduce the learning curve of the system by finding similarity amongst user profiles using a number of similarity measurement techniques. One of these techniques is User

Cluster Assignment. In this method, based on user's explicit feedback, users decide under which specific cluster they want to be. This decision will be made after they are informed about the contents of those clusters. By the loans and reservation pattern which is applied in the library, not only system can understand users' needs but also allows users with similar behaviors to share information with each other.

4.5 Dynamic Collaborative Profiling

Dynamic collaborative profiling is the final approach in which users with similar behaviour are clustered into peer groups based on their profiles. Information will be filtered based on the group interests. System Cluster Assignment technique is used for this approach. Although the process is similar to 2.2.4, the clustering technique is different. It is based on dynamic feedback via the loans and reservation pattern.

4.6 Implicit User Interest Capturing

The aim of the approach presented in the [15] is to develop a method to capture the searchers needs according to their behavior without asking them to give feedback explicitly. Most systems that are based on relevance feedback force users to choose between a binary choice of relevant or not relevant for the returned results. This makes users uncomfortable to make a decision on what document to assess as relevant or not relevant. Our system omitted these difficulties by implicitly monitoring the interaction of the user with the results [15].

In [15] the authors also measure the degree of potential changes in user information needs and tailor the results according to that level. If the degree of change is great, a new search result will be returned, while, if the degree is minor, a re-ordering or re-ranking for the retrieved list will be sufficient.

5. CONTEXT AWARE RSS AGGREGATOR

In [6] a context-aware approach to increase the precision in the information aggregator is illustrated. In each RSS feed there are some contexts available. By using ACM category classification or generic on-line dictionaries, system agent will be able to understand the RSS context and categorize them. The presented architecture consists of four main parts, namely ontology, content semantics parser, context-aware agents, and context description model. The context description model consists of service description and content description. The service description consists of context schema and service context model. Context schema talks about the schema and basic vocabulary of the context. The service context holds basic information about the service such as name, type, and version. The content description is the concrete part because it deals with the real content of the resource objects [6].

The most important point in this paper [6] is that the approach is based on the semantics of the articles rather than the user interest. In other words, the user interest is not considered here. The filtering is based on the semantically matching items with the user query.

6. PROPOSED WORK

As the popularity of "weblogs" (or blogs) has been growing over time and as many users have been closely following the new postings of their favorite blogs, there has been a dramatic increase in the use of XML data to deliver information over the Web. In particular, personal weblogs, news Websites, and discussion forums are now delivering up-to-date postings to their subscribers using the RSS protocol. To help users access new content in this RSS domain, a number of RSS aggregation services and blog search engines have appeared recently and are gaining popularity. Using these services, a user can either (1) specify the set of RSS sources that she is interested in, so that the user is notified whenever new content appears at the sources (either through email or when the user logs into the service) or (2) conduct a keyword-based search to retrieve all content containing the keyword. Clearly, having a central access point makes it significantly simpler to discover and access new content from a large number of diverse RSS sources.

The aggregation can be done either at a desktop (e.g., RSS-feed readers) or at a central server (e.g., Blog lines or Google Reader). This problem is similar to the index refresh problem for Web-search engines, but two important properties of the information in the RSS domain make this problem unique and interesting:

• The information in the RSS domain is often time-sensitive. Most new RSS content is related to current world events, so its value and significance deteriorates rapidly as time passes. An effective RSS aggregator, therefore, has to retrieve new content quickly and make it available to its users close to real time. This requirement is in contrast to general Web search engines where the temporal requirement is not as strict. For example, it is often acceptable to index a new Webpage within, say, a month of its creation for the majority of Webpages.

• For general search engines, users mainly focus on the quality of the returned pages and largely ignore what is not returned. Based on this observation, researchers have argued for and mainly focused on improving the quality of the top k result, and the page-refresh policies have also been designed to improve the freshness of the top-ranked pages. For RSS feeds, however, many users often have a set of their favorite sources and are particularly interested in reading the new content from these sources. Therefore, users do notice if the new content from their favorite sources is missing from the aggregator. The time-sensitivity of the RSS domain fundamentally changes how one should model the generation of new content in this domain and makes it necessary to design a new content-monitoring policy.

6.1 Framework

An online RSS aggregator, which is a distributed information system that consists of n data sources, a single aggregator and a number of subscribers, constantly generate new pieces of information referred to as new postings. A subscriber, in turn, consumes new postings from the aggregator. There does exist another push-based architecture where data sources notify ping servers through an XML-RPC protocol whenever there is a new posting. Upon receiving such messages, the aggregator then decides when to retrieve new postings.

7. RELATED WORK

Web crawling is a well-studied research problem. Various researches have investigated the problem of maintaining a fresh copy of Webpages for search engines. While the general problem is similar, the exact model and overall goals are significantly different from ours. Some papers [16, 18] investigated the problem of minimizing the time to download one snapshot of the Web by efficiently distributing the downloading task to multiple distributed processes. In more recent work [24-27], researchers have proposed new crawling strategies to improve the user satisfaction for Web search engines by using more sophisticated goal metrics that incorporate the query load and the user click-through data. Since this body of work mainly focuses on getting improvement by exploiting the user behavior in the context of a Web search, it still assumes a relatively simple model to predict the changes of Webpages.

In terms of improving the user's browsing experience in the time perspective, pre-fetching is a technique commonly used to reduce the wait-time of loading Webpages. Such a technique can be deployed at different locations on the Web. In particular, when deployed on the client side, pre-fetching algorithms predicts the links on the current page that are likely to be accessed by the user in the future and requests them in advance; hence, it reduces the waiting time of the user when loading pages. In the case of deployment on the server-side, pre-fetching works by analyzing the Web access log for document access patterns; it then pre-fetches subsequent documents from disk into main memory to reduce the access time when they are requested by the clients afterwards.

Similar techniques can also be deployed on proxy servers that serve a collection of users within a subnet. In the context of a relational database, the use of periodic inhomogeneous Poisson process (referred to as Recurrent Piecewise-Constant Poisson process in the reference) to model record updates in a database. Due to the difference in the general goal and user requirements, however, its overall problem formulation and final solutions are significantly different from ours.

While these push-based approaches have significant benefits, whether they will be widely adopted for the general Web remains to be seen because it requires additional administrative work for publishers to adopt such scheme. There have been recent efforts to make Web crawling more efficient by improving the underlying protocol. For example, Google sitemap protocol allows a site administrator to publish the list of pages available at her site at a predefined location together with the last modification dates of the pages. While this new protocol helps a crawler discover new Webpages and their changes more efficiently, it is still based on the pull architecture, where a Web crawler is still responsible for periodically contacting the sites and downloading changes. Therefore, even if this protocol is widely deployed, the proposed monitoring policy will still be helpful in reducing the retrieval delay of new postings.

Researchers have studied publisher-subscriber systems and proposed strategies for the efficient dissemination of information in these systems. This body of work mainly focuses on the efficient filtering of the overwhelming incoming data stream against a large pool of existing subscriber profiles and on the efficient data delivery method in the Internet scale; different from this body of work, our aggregator is not passively waiting for new data to come in; instead, the aggregator monitors and actively pulls from different data sources to collect new postings.

8. CONCLUSION

With the growing popularity of "Web 2.0" services, it has become easy and convenient for less technically savvy users to publish content on the Web, leading to the explosion of user-generated online content. However, such an increase in quantity does not guarantee an improvement in the quality. Oftentimes, users find themselves overwhelmed by the magnitude of new information being generated every day.

Researchers and engineers have been building online content aggregators that help users better manage their subscribed RSS feeds and tap into Web 2.0 information easily. Such aggregators, when supporting a large number of users with diverse interests and subscriptions, face many challenges and opportunities such as: delivering updated content, providing good personalization efficiently and accurately, and data mining user generated content for improving the system. In this dissertation, through an analysis of existing online content aggregators, we have studied the challenges and opportunities in delivering fresh and personalized content to users and how to make use of the data collected from users' interactions with the systems.

9. REFERENCES

[1] R.T. Freeman: Web Document Search, Organisation and Exploration Using Self-Organising Neural Networks, PhD Thesis, Faculty of Engineering and Physical Sciences, School of Electrical & Electronic Engineering, University of Manchester: Manchester (2004)

[2] A. Qamra, B. Tseng, E.Y. Chang: Mining blog stories using community-based and temporal clustering. In: 15th ACM international conference on Information and knowledge management CIKM, pp – 58-67, ACM (2006)

[3] G. Paliouras, M. Alexandros, C. Ntoutsis, A. Alexopoulos, and C. Skourlas: PNS: Personalized Multi-Source News Delivery. In: 10th International Conference on Knowledge-Based & Intelligent Information & Engineering Systems. LNCS, vol. 4252, pp. 1152 – 1161. Springer (2006)

[4] X. Li, J. Yan, Z-H. Deng, L. Ji, W. Fan, B. Zhang, Z. Chen: A Novel Clustering-based RSS Aggregator, In: 16th international conference on World Wide Web, pp. 1309 – 1310. ACM (2007)

[5] N. Agarwal, M. Galan, H. Liu, S. Subramanya: Clustering Blogs with Collective Wisdom, In: 8th International Conference on Web Engineering, ICWE'08. pp. 336 – 339. IEEE (2008)

[6] W. Huang, and D. Webster: Enabling Context-Aware Agents to Understand Semantic Resources on The WWW and The SemanticWeb, International Conference on Web Intelligence (WI'04), pp. 138 – 144. IEEE (2004)

[7] D. Webster, W. Huang, D. Mundy, P. Warren, Context-Orientated News Filtering for Web 2.0 and Beyond, In: 15th International World Wide Web Conference, pp. 1001 – 1002. ACM (2006)

[8] M. Thelwall, R. Prabowo: Identifying and Characterizing Public Science-Related Fears From RSS Feeds, Journal of the American Society for Information Science and Technology, 58(3), 379-390 (2007)

[9] T. Kohonen: Self-Organizing Maps, Third Extended Edition, Springer (2001)

[10] R.T. Freeman: Topological Tree Clustering of Web Search Results. In Intelligent Data Engineering and Automated Learning, IDEAL 2006, LNCS, vol. 4224, pp. 789-797, Springer (2006)

[11] R.T. Freeman: Topological Tree Clustering of Social Network Search Results. In: Intelligent Data Engineering and Automated Learning - IDEAL 2007. LNCS, vol. 4481, pp. 760-769. Springer (2007)

[12] G. Salton: Automatic text processing - the transformation, analysis, and retrieval of information by computer. Addison-Wesley (1989)

[13] R.T. Freeman, and H. Yin: Web content management by self-organization. Neural Networks, IEEE Transactions on. 16(5), 1256-1268 (2005)

[14] R.T. Freeman and H. Yin.: Adaptive topological tree structure for document organisation and visualisation. Neural Networks. 17(8-9), 1255-1271 (2004)