Topic Initiator Detection on the World Wide Web*

Xin Jin
University of Illinois at Urbana-Champaign
201 N. Goodwin Ave.
Urbana, IL USA
xinjin3@illinois.edu

Scott Spangler[1], Rui Ma[2]
[1]IBM Almaden Research Center
[2]IBM China Research Lab
[1]spangles@almaden.ibm.com
[2]maruicrl@cn.ibm.com

Jiawei Han
University of Illinois at Urbana-Champaign
201 N. Goodwin Ave.
Urbana, IL USA
hanj@cs.uiuc.edu

ABSTRACT

In this paper we introduce a new Web mining and search technique - Topic Initiator Detection (TID) on the Web. Given a topic query on the Internet and the resulting collection of time-stamped web documents which contain the query keywords, the task of TID is to automatically return which web document (or its author) initiated the topic or was the first to discuss about the topic.

To deal with the TID problem, we design a system framework and propose algorithm InitRank (**Init**iator **Rank**ing) to rank the web documents by their possibility to be the topic initiator. We first extract features from the web documents and design several topic initiator indicators. Then, we propose a TCL graph which integrates the Time, Content and Link information and design an optimization framework over the graph to compute InitRank. Experiments show that compared with baseline methods, such as direct time sorting, well-known link based ranking algorithms PageRank and HITS, InitRank achieves the best overall performance with high effectiveness and robustness. In case studies, we successfully detected (1) the first web document related to a famous rumor of an Australia product banned in USA and (2) the pre-release of IBM and Google Cloud Computing collaboration before the official announcement.

Categories and Subject Descriptors

H.3.3 [**Information Search & Retrieval**]: Retrieval models; H.2.8 [**Information Systems**]: Data Mining

General Terms

Algorithms, Experimentation

Keywords

Web mining, ranking, information retrieval, topic initiator

1. INTRODUCTION

In many cases on the Web, given a topic query, we want to know which web document (or its author) is the one to initiate the topic or the first one to talk about the topic. For example, someone started a rumor about a product on the Web and generated a lot of discussions on this topic. The company would like to know who started this rumor. Based on our knowledge, there is no current system supports this technique or service.

General search engines, such as Google, Yahoo and Bing, only return webpages which are most relevant to the query. For example, Google Web Search supports searching by query and returns webpages sorted by the PageRank based relevancy scores. This method cannot find the topic initiator.

Google News is a service that automatically clusters new articles into groups, each of which contains articles for the same topic, and provides sorting based on relevance or date. The problem is that the clustering results are not always correct, in some cases the articles in the same group are not about the same topic. For example, there was a news group in the Sci/Tech section which contained 111 news articles mostly about Microsoft's next version of Internet Explorer IE 8. Within the cluster, the first result was titled "Microsoft IE8 To Make Stealth Surfing Easier". Unfortunately, the group was wrong even for the second article titled "Mozilla steps up Firefox 3 push", which discussed Firefox 3.

Some specific search engines provide searching by query and sorting the search results by dates. Google News Search as a good example. However, it only supports search for a query and simply sort the results by dates. We will show later that only using date information is far from enough. In addition, Google News Search only supports news articles, but our framework works for the whole Internet by integrating all the information on the web, including the news, blogs, forums, newsgroups, etc.

Another drawback of existing systems is that they only support webpage level analysis. However, we go deeper into the web document level. When a user wants to find which web document is the initiator, all the three major search engines do not work to this level of details.

In this paper we introduce a new web mining and search engine technique/service - Topic Initiator Detection (TID) on the Web. Given a topic query, the system finds all webpages containing the query word/words. Then it extracts the web documents within each webpage, examples of web documents are news articles, blogs, forums and newsgroup posts. The difference between a web document and a webpage is that a webpage may contain more than one web document. Several web documents may appear on the same webpage. For example, blog articles could be posted on the same blog page. Based on web documents, information

*This research was supported by IBM and also sponsored in part by the U.S. National Science Foundation under grants IIS-08-42769 and IIS-09-05215.

WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

such as the author name, time, content and links are extracted. Finally, the system returns a list of web documents (together with the author names) ranked by their possibility to be topic initiator or be the first to talk about the topic.

We give a formal **definition** for problem of Topic Initiator Detection (TID) on Web as follows,

Input: Given a topic query q and a collection $D(q)$ of web documents which contain the query word/words. Suppose D(q) consists of N time-stamped web documents, $D(q) = \{d_1, d_2, , d_N\}$ and the associated time information is $T = \{t_1, t_2, , t_N\}$, where d_i denotes web document $i(i = 1, \ldots, N)$ and t_i represents its time stamp.

Output: The web document which initiated the topic or was the first to talk about the topic.

An **intuitive solution** for the TID problem is as follows: (1) according to the topic query, return all web documents that contain the query; (2) sort the documents by time; and (3) select the first one as the initiator. The performance of this intuitive method is poor, because a web document that appears early may just happen to contain the query words, but does not really talk about the topic.

Another style of method is to use link-based algorithms, such as InDegree, PageRank [16] and HITS [5], to choose the one with the highest ranking score as the topic initiator. However, the true topic initiator may only have a small number of citations or even not be cited by any other web documents, and a following article that appears in a popular website may get a lot of citations and obtain the highest ranking score by link.

We develop InitRank to rank the possibility of a web document to be the topic initiator, based on time, content, link and some other useful information. We first introduce several topic initiator indicators, and then propose a TCL graph which integrates Time-Content-Link information and design an optimization framework to compute InitRank.

Contributions of this paper are:

1. Introduce a new web mining and searching technique or service - Topic Initiator Detection (TID) on the Web. Given a topic query, return which web document (or its author) initiated the topic or was the first to discuss about the topic.

2. Design a system framework for TID on the Web and propose algorithm InitRank to automatically find the topic initiator. Based on a ranking score initialization using initiator indicators, InitRank refines the score within a optimization framework over a TCL graph.

3. Experiments on real datasets show the good performance of our algorithm and verify its effectiveness and robustness.

4. Show interesting findings with case studies.

2. RELATED WORK

The most related work is the research on New Event Detection (NED) [22], which is also called Novelty Detection or First Story Detection (FSD) [23]. The task of NED is to automatically detect the earliest report for each event as soon as that report arrives in the sequence of documents. NED is the most difficult task in the research area of Topic Detection and Tracking (TDT) [1], which is an important research area in Web Mining [12]. Most NED systems basically work by comparing a document to all the documents in the past, and use a threshold on the similarity scores to detect novel stories. If all the similarity scores are below the predefined threshold, the document is predicted as the first story of a novel event [22].

We give a brief description of several systems for NED as mentioned in [6]. The UMass (Univ. of Massachusetts) method performs clustering – implementing a modified version of the single-pass clustering algorithm – on the set of time-stamped documents and returns the first document in each cluster as the result. The CMU approach represents document using vector space model with term weighting and uses single-pass clustering algorithm to partition stories into different topic groups. The general idea is similar to the UMass method. The UPenn (University of Pennsylvania) approach begins with clusters of size one and merges similar clusters. Stories are compared to the preceding ones and merge their clusters when the similarity is high enough. If a story cannot be combined with any other existing cluster, it becomes a new cluster, thus the story is a new story.

The above systems do not work for TID because there are no multiple clusters, and, most importantly, they do not go deeper to solve the problem of how to select the right first story within a cluster.

The approach proposed in [6] uses TF-IDF term weighting and assigns FSD-value to each story according to some rules. It works sequentially: (1) the first story in the collection is always a first-story (FAD-value = 0), (2) the second story is evaluated by calculating a measurement of similarity based on the occurrences of terms that were in the previous story, and (3) continue these steps for each subsequent story, the FSD-value will be lower if the story contains a large number of previously unknown terms. A story is identified as a first-story if its FSD-value is under a threshold value. This method does not work for TID because the first story in time will always have the lowest FSD-value and be identified as the first story, which, as we have already discussed, is incorrect in many cases.

In paper [23], the authors propose a two-level approach for novelty detection: (1) using a supervised learning algorithm to classify the on-line document stream into predefined broad topic categories, and (2) performing topic-conditioned novelty detection for documents in each topic. The limitation of the approach is that it needs training data for classification, thus it is a supervised method. It is not applicable to the TID problem which is unsupervised.

Additional differences between TID and NED are: (1) NED works sequentially, but TID is not required to work sequentially and thus be more flexible. (2) TID is web based and contains other related information, aside from only time and text.

3. SYSTEM FRAMEWORK

We present a framework system for Topic Initiator Detection on Web. The goal is that given a topic query, we want to know which web document is the one to initiate the topic or the first one to talk about the topic. Note that it's a search service that traditional search engines, such as Google, Yahoo and Bing Search, do not support.

As shown in Figure 1, the general system framework works as follows: (1) beginning with the user submitted topic query, fetch webpages that contain the query keywords; (2)

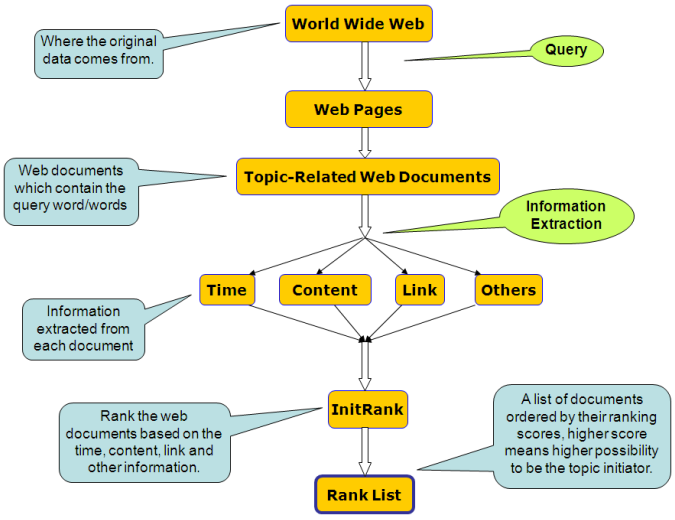

Figure 1: Framework for Topic Initiator Detection on the World Wide Web.

extract the web documents from the webpages; (3) extract information such as author name, date, content and links, for forums, we also extract the forum id and thread id; and (4) perform topic initiator ranking algorithm InitRank and return a sorted list. Note that step (1) can be performed efficiently based on inverted index, steps (2) and (3) can be pre-computed.

Data Source. Since our goal is to mine the information from the whole Web, only one type of data source is not enough. An initiator may come from a blog website, a news website or even from a newsgroup discussion. So in our framework, the mining process is based on data sources in the whole Web which consists of a lot of different types of information, such as blogs, news and newsgroups.

Web Document v.s. Webpage. Our analysis is based on web document level instead of webpage level. Web documents are extracted from the webpages. Each webpage may contain one or multiple web documents. Some webpages even contain less than one web document. For example, some news websites divide an article into several webpages to gain more clicks. The webpage and web document mapping describes the relationship between a webpage and a web document. There are three kinds of Webpage-WebDocument mapping: One-One, One-Multiple and Multiple-One. One-One maps one webpage to one web document, One-Multiple maps one webpage to multiple web documents. For example, a blog page may contain multiple posts. Multiple-One maps multiple webpages to one web document.

Information Extraction. For each web document, our system extracts many related information, as listed in Table 1. The attribute Date is the time the web document was published online, not the date entity, if there is any, extracted from the document content.

Document Preprocessing. The web documents go through the following preprocessors: stopword-removal [4], synonymy, stemming [18] [19] and noise-removal. Words such as "cdata", "nbsp", "http", "www", "pdf" and "html", are added to the standard stop-word list, because they are common in many webpage documents and provide little information about the topic. To handle synonyms, such as "USA" with "U.S.", we

Table 1: Major Attributes Extracted from a Web Document

Attribute	Description
Date	Publication time.
Domain	Website domain of the web document
URL	URL of the web document
Title	Title of the web document
Text	Text content of the web document
ThreadId	Identification of the thread
ForumId	Identification of the forum
ForumName	Name of the forum
Author	Name of the author
BBStype	Type of BBS (Bulletin Board System)
SourceType	MessageBoard, Blog, News, etc.
Country	Country of the website
LinkURL	Set of link/citation URLąś
LinkDomain	Set of link/citation website domains
Query	Keyword(s) of the query

give a list of synonyms and transform those different words to a single form. There are a lot of noisy terms on the web and many of them only appear in very few number of web documents, we remove terms with very low document frequency, e.g. those appearing in less than 2 web documents.

Document Similarity. We use vector space model to represent a web document and adopt the standard $tf * idf$ weighting method. The weight of a term is computed as $tf * (1 + log(N/df))$, where df is the number of web document the term appears in. Other weighting schemes can also be used, such as PN [21] and BM 25 [20]. To estimate $Sim(d_i, d_j)$, the content relationship between two web documents d_i and d_j, we use the cosine (normalized dot product) similarity measure.

4. RANK THE INITIATOR

This section presents algorithm to rank the web documents for TID on the Web. We first define normalization functions, and then introduce several basic topic initiator indicators, the final ranking schemes (combo ranking and graph-based refinement) in detail.

Based on the sigmoid function, we design a normalization function $SNInc()$ as follows,

$$SNInc(x) = \frac{2}{1 + e^{-x/\mu}} - 1 \quad (1)$$

$SNInc(x)$ is a normalized increasing function. When $x > 0$, the value of $SNInc(x)$ ranges from 0 to 1.

A good property of this function is that small x has higher impact on the score, while big x has lower impact on the score. Parameter μ controls what the function curve looks like. The setting of μ allows us to decide where we want the change of x has little impact on the change of score. For example, if $\mu = 5$, when x is bigger than around 50, the score will be close to 1. The $SNDec()$ function is defined as $SNDec(x) = 1 - SNInc(x)$, which is a normalized decreasing function.

4.1 Indicators for Ranking the Initiator

This section presents several indicators for ranking the possibility of a web document to be the topic initiator.

4.1.1 Centrality

A topic initiator starts a topic and spreads the information via many following web documents, so it should be located around the content center, i.e., similar in content with its followers. Thus the similarities between a web document and all other web documents give us a hint on the potential of the web document to be the topic initiator.

To estimate the *Centrality* of a web document d_i, there are two types of measures: $AverSim$ and $CenterSim$. We define $AverSim$ as the average similarity between d_i and all other web documents in the query result list. $CenterSim$ is defined as the similarity between d_i and the center d_c of all the web documents. For vector space model, the center is the mean point of the document points; for language model, the center is background model estimated from the documents.

$$AverSim(d_i) = \frac{1}{N-1}\sum_{j\neq i}^{N} Sim(d_i, d_j) \quad (2)$$

$$CenterSim(d_i) = Sim(d_i, d_c) \quad (3)$$

N is the size of the set of web documents containing the query. The computational complexity of $CenterSim$ and $AverSim$ is $O(N)$ and $O(N^2)$, respectively. So $CenterSim$ is more efficient.

Note that achieving the highest *Centrality* score does not necessarily mean the web document is the topic initiator, because a following web document may contain more detailed information about the topic, and has higher *Centrality* score than the topic initiator.

4.1.2 Novelty

Since the topic initiator is the beginning of the topic, it should be novel. More specifically, the topic initiator should not only be similar to its following web documents, but also dissimilar to its earlier web documents. This leads to two factors which consider both time and content information:

$$AS_L(d_i) = \frac{1}{N_L}\sum_{t_j > t_i} Sim(d_i, d_j) \quad (4)$$

$$AS_EMax(d_i) = argmax_{t_j<t_i}\{Sim(d_i, d_j)\} \quad (5)$$

where N_L is the number of web documents that appear later than web document d_i.

$AS_L(d_i)$ is the average similarity between d_i and its later web documents, while $AS_EMax(d_i)$ is the highest similarity between d_i and its earlier web documents. We want $AS_L(d_i)$ to be high and $AS_EMax(d_i)$ to be low, so the *Novelty* of a web document d_i is defined as follows,

$$NOVE(d_i) = \frac{AS_L(d_i) - AS_EMax(d_i) + 1}{2} \quad (6)$$

This function is designed to range within $[0, 1]$, since $AS_L(d_i)$ and $AS_EMax(d_i)$ are independent and both range from 0 to 1.

4.1.3 Originality

The originality factor is introduced because a topic initiator should be original. We consider the following rules to decide whether the web document is original or not:

Rule 1. If the title of a post begins with "Re:" or other reply indicators, such as "RE:", "Reply #99 on:" and "reply to why girls don't like drugs", we consider the post as not original. The possibility is low for the post to be a topic initiator, because it is unusual that someone starts a new burst of topic when reply to a topic post.

Rule 2. For posts within the same thread of the same forum, we consider those not posted in the first day as not original. We form a new attribute ThrForId, which is a merging of the ThreadId and ForumId. In most cases, web documents which share the same ThrForId belong to the same group of discussion, and thus only the web documents from the first day are considered to be original.

Rule 3. The problem for rule 2 is that even on the same day, there could be many posts. Ideally the first post should be chosen, because all others are just replies. However, if we do not know the exact time of each post, we simply decide they are all original.

Based on originality (ORIG) information, the possibility for the web document d_i to be the topic initiator is evaluated as follows

$$ORIG(d_i) = \begin{cases} 1 & original \\ \theta & not\ \ original \end{cases} \quad (7)$$

Parameter θ ($\theta \in [0, 1]$) controls the possibility of a non-original web document to be the initiator. For simplicity, we set $\theta = 0$ to ignore any non-original web documents. In this case, the originality factor works as a filtering function.

4.1.4 Document Length Factor (DLF)

Some forum or newsgroup posts are very short, but contain a lot of query keywords, and thus have high overall similarity to other web documents. To deal with this problem, we make the assumption that a web document should be long enough to provide useful information.

Let $L(d_i)$ as the length (number of words) of web document d_i, we define Document Length Factor (DLF) to utilize the above assumption. DLF is computed as a normalized score based the document length using the $SNInc()$ function.

$$DLF(d_i) = SNInc(L(d_i)) \quad (8)$$

The length of a web document usually ranges from 1 to over 3000 words, and the average length among our dataset is about 50 words. We thus assume that a web document which contains more than around 50 words brings enough information to start a wildly spread topic. Based on the property of the $SNInc()$ function, $\mu = 7$ is a good setting for our task. The reason is that under this setting, web documents which are longer than around 50 will have a DLF score close to 1, and thus a web document with 50 words length has similar DLF score with those with 500 words, because they are all long enough. Meanwhile, a web document with very few words, e.g. 5, will have a very small DLF score, which indicates that the web document is too short to be a topic initiator.

4.1.5 Term Allocation Compactness (TAC)

Distance between term occurrences has been shown to be useful for relevance weighting in retrieval [9]. Term gap gives a hint on the topic focus of the web document and we are especially interested in the query terms in the web document. If the query terms appear close to each other in a web document, it is more confident to say that the web document is about the query topic. Otherwise, if they appear far away from each other, the possibility is low for the web document

to be focusing on the query topic. We introduce the Term Allocation Compactness (TAC) score to utilize the term gap for ranking.

A term may appear in the document d for multiple times. For a query of n terms, let q_i denotes the ith $(i = 1, \ldots, n)$ term of the query, m_i denotes the number of appearances of term q_i in the document d, $Z_i = 1, \ldots, m_i$, l_{ij} denotes the location of the jth $(j \in Z_i)$ appearance of term q_i in the document. The value of l_{ij} ranges from 1 to L, and L is the length of the document. Define c as a combination of the locations of the terms in the document.

$$c = \{l_{1j_1}, l_{1j_1}, \ldots, l_{nj_n} | j_i \in Z_i\} \quad (9)$$

Denote $C = \{c\}$ as the set of combinations for the query in the document, and M as the number of different combinations,

$$M = |C| = \prod_{i=1}^{n} m_i \quad (10)$$

We only consider absolute gap between terms, and ignore the relative order. For example, "Google and IBM" is considered as the same as "IBM and Google". To facilitate computation, the locations in c are sorted in increasing order. Then c is re-represented cs,

$$cs = \{ls_1, ls_2, \ldots, ls_n\} \quad (11)$$

When ls_i is the location of the ith term in the sorted cs. Based on sorted combination cs, the average gap between terms is calculated as follows,

$$AveGap(cs) = \frac{1}{n-1} \sum_{i=1}^{n-1} (ls_{i+1} - ls_i - 1) \quad (12)$$

Select the combination with the minimum average gap,

$$MinGap(d) = argmin_{cs \in C}\{AveGap(cs)\} \quad (13)$$

Finally, TAC is calculated as a normalized score,

$$TAC(d) = SNDec(MinGap(d)) \quad (14)$$

Note that although term gap is a good topic indicator, it does not necessarily mean a web document with compact query terms allocation is definitely talking about the topic. We still have to check the whole content of the document to see its true topic focus.

4.1.6 Earliness

Intuitively, given the topic query and the web documents containing the query keyword(s), a web document which appears earlier should have higher possibility to be the topic initiator.

Based on this assumption, a naive approach to the estimate the possibility for the web document d_i to be the topic initiator is the following ranking function of the time information.

$$P_{Time}(d_i) = \frac{T_{End} - t_i}{T_{End} - T_{Begin}} \quad (15)$$

where, $T_{Begin} = min\{t_i\}$ and $T_{End} = max\{t_i\}$.

There are two major disadvantages of the naive approach: (1) if there is a noisy web document whose publication date is much earlier compared with other web documents, it will dominate the ranking function and make all other web documents have similar scores; and (2) all publication dates are equally important, however, if the web documents in the same date are all outliers (or non-relevant), this date should not account much.

We propose a better method which considers both the time order and the content of the web documents within the same date. The basic idea is to use time order instead of exact time gap for solving problem (1) and use content analysis to give relevance score/weight to the dates for solving problem (2).

We sort the dates in increasing order $O = \{st_1, st_2, \ldots, st_P\}$, P is the number of distinct dates ($P \leq N$, where N is the number of web documents). Define the order of time/date t as $Order(t) = q$, where $t = st_q$. Since st_j is the jth sorted date, $Order(st_j) = j$.

For a date st_j, let $D(st_j) = \{d_i | t_i = st_j\}$ as the set of web documents whose publication date is st_j, We define MCS as the maximum content score of those documents,

$$MCS(st_j) = argmax_{d_j \in D(st_j)}\{ContentScore(d_j)\} \quad (16)$$

For simplicity, $Centrality$ is used as the ContentScore.

For a date st_j, define its importance/weight $W(st_j)$ as a score related to the $Order$ and MCS, and normalize the $Order$ using the $SNDec()$ function,

$$W(st_j) = SNDec(Order(st_j)) * MCS(st_j) \quad (17)$$

Finally, we get the ranking for web document d_i by earliness (EARL) as follows,

$$EARL(d_i) = \frac{\sum_{j=1}^{Order(t_i)} W(st_j)}{\sum_{j=1}^{P} W(st_j)} \quad (18)$$

The limitation of directly using time information is that the first web document is not necessarily the topic initiator. Because it may happen to contain these query words, but is not really talking about the topic. Even if we consider weighting by the order and content, the current ranking function will still rank the first document as top 1. We still need other factors to get the true topic initiator.

4.2 Rank Scheme 1 - ComboRank

Using only time, originality, content or link in isolation gives poor performance. If we only use originality, there could be a lot of original web documents. If we only consider time, there could be a lot of web documents ranking high but not really talking about the query topic. If we only consider content similarity, the topic initiator is not necessarily the web document with the highest overall similarity with other web documents, because it is possible that some following web documents contain more information about the topic, and thus have higher overall similarity with other web documents.

We propose our first scheme for ranking the topic initiator. Assume the indicators, such as originality, content similarity, term gap and web document length are factors independent of each other. Then the topic initiator can be ranked as a multiplicative model of the basic indicators,

$$ComboRank = \prod D_i \quad (19)$$

where $D_i's$ are the indicators, such as ORIG, DLF, TAC, EARL, LINK and CenterSum, where LINK is the normalized InDegree. We call this approach $ComboRank$, such a

combined solution shows stronger performance than individual indicators in a robust fashion in diverse situations.

4.3 Rank Scheme 2 - InitRank with Graph based Refinement

One disadvantage of ComboRank is that there are many components, a big error in a component may have big impact on the final score. To solve the problem, we propose our second approach, called InitRank (**Init**iator **Rank**ing), to smooth the scores based on graph based refinement.

We first use some basic indicators to get ranking score initialization for each web document, for example,

$$r^{\dagger} = ORIG * DLF * TAC \tag{20}$$

Then refine the scores based on a TCL graph model as described below.

4.3.1 TCL graph

We propose TCL graph to integrate the time(T), content(C) and link(L) information. Each node denotes a web document. Add two kinds of directed edges: one type is based on the link information; another type represents the semantic relationship and *information flow*, the relationship confidence is based on the content similarity and the information flow direction is based on the time order. The second type is used to capture the situation, as found in many real cases, where the information of a web document come from another one but it does not link to it. Based on content similarity, we can estimate such semantic relationship.

Figure 2 shows an example TCL graph simplified from a real query result about "Vegemite ban". It shows 13 web documents ordered in four dates. The solid directed edge is the link between two web documents. The dashed directed edge shows the hidden semantic relationship between two web documents. The weight indicates the confidence of such relationship. The direction represents the information flow from one node to another. We make the direction goes from the web document which appears later to the one which appears earlier, in order to show that the information of the later web document can be traced back to the earlier web document. For two web documents in the same date, it's hard to distinguish the information sender and receiver. So we assume both directions are possible.

Web document 3 is the true topic initiator. Web document 1 appears on the first date, but does not really talk about the topic indicated by the query, even though it contains the query keywords (We will show how this could happen in our case study in the experiments part of the paper). So the method solely based on time ordering will falsely rank 1 as the topic initiator. Web document 6 has the highest overall similarity to other web documents; however, it is not the original topic initiator. Web document 7 gets the biggest number of inlinks but is not the initiator. In order to find the true topic initiator, we need to integrate the link and semantic relationship.

Definition. In a TCL graph, V is the set of vertices/nodes, $e_{ij} =< i, j >$ is a directed edge from node i to node j, E is the set of edges in this graph. E_L is the set of edges formed by the link(L) information and E_S is the set of edges modeling the semantic(S) relationship between nodes, w_{ij} is the weight/confidence of the relationship estimated by the content similarity, the direction goes from web document i to j and i appears no earlier than j.

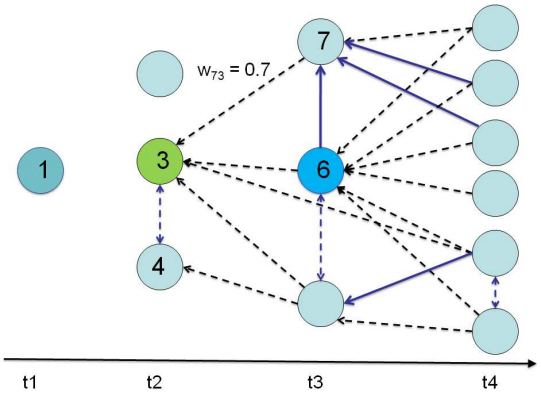

Figure 2: An example TCL graph.

4.3.2 Optimization over the Graph

Denote Φ_i as the influence or importance of node i, we can estimate a node's influence by the number of nodes which link to it and the number of nodes which shows hidden semantic relationship with it. If the node has many inlinked nodes or semantic related nodes, its importance should be high.

Let $R = \{r_i\}(i = 1, ..., |V|)$, r_i is the initiator ranking of node i, optimize the following objective function $O(R)$,

$$\begin{aligned} O(R) &= \alpha \sum_{i \in V} \Psi_i || \frac{r_i}{\Psi_i} - \frac{r_i^{\dagger}}{\Psi_i} ||^2 \\ &+ \beta \sum_{<j,i> \in E_L} || \frac{r_i}{\Psi_i} - \frac{r_j}{N_j^O} ||^2 \\ &+ \gamma \sum_{<j,i> \in E_S} w_{ij} || \frac{r_i}{\Psi_i} - \frac{r_j}{W_j^O} ||^2 \end{aligned} \tag{21}$$

$$\tag{22}$$

where $W_j^O = \sum_{<j,i> \in E_S} w_{ji}$ is the sum of the weights for the edges between node j and the nodes that j directs to, N_j^O is the number of nodes that node j links to.

Among the three components in the objective function, the first component means that the refined score should not deviate too much from the initialization score, we use $||r_i/\Psi_i - r_i^{\dagger}/\Psi_i||^2$ instead of $||r_i - r_i^{\dagger}||^2$ in order to be comparable to the other two components; the second component means the semantic information sent out from the initiator is similar to the information received; the third component means similar idea as the second terms but such flow is indicated by link information.

Our goal is to find $R = R^*$ to minimize the cost function, $R^* = argmin\{O(R)\}$. R^* is the final ranking score in our InitRank algorithm. To minimize $O(R)$, we compute its first-order partial derivatives,

$$\begin{aligned} \frac{\partial O(R)}{\partial r_i} &= 2\frac{\alpha}{\Psi_i}(r_i - r_i^{\dagger}) \\ &+ 2\frac{\beta}{\Psi_i} \sum_{j \in V_i^L} (\frac{r_i}{\Psi_i} - \frac{r_j}{N_j^O}) \\ &+ 2\frac{\gamma}{\Psi_i} \sum_{j \in V_i^S} (\frac{w_{ij} r_i}{\Psi_i} - \frac{w_{ij} r_j}{W_j^O}) \end{aligned} \tag{23}$$

where V_i^L is the set of nodes which link to node i, V_i^S is the set of nodes which have semantic link to node i.

Let $\frac{\partial O(R)}{\partial r_i} = 0$, we get

$$\begin{aligned} r_i &= \frac{\alpha}{\alpha + \beta\frac{N_i^I}{\Psi_i} + \gamma\frac{\sum_{j\in V_i^S} w_{ij}}{\Psi_i}} r_i^\dagger \\ &+ \frac{\beta}{\alpha + \beta\frac{N_i^I}{\Psi_i} + \gamma\frac{\sum_{j\in V_i^S} w_{ij}}{\Psi_i}} \sum_{<j,i>\in E_L} \frac{1}{N_j^O} r_j \\ &+ \frac{\gamma}{\alpha + \beta\frac{N_i^I}{\Psi_i} + \gamma\frac{\sum_{j\in V_i^S} w_{ij}}{\Psi_i}} \sum_{<j,i>\in E_S} \frac{w_{ij}}{W_j^O} r_j \end{aligned} \quad (24)$$

where $N_i^I = |V_i^L|$.

R is initialized as $\{r_i^\dagger\}$, the final score R^* is obtained by iteratively updating all r_i via Equation 24.

We could put a window which only include semantic links within such window at each time position to avoid building too big TCL graph.

Connection to Absorption Random Walk. Equation 24 can be understood as an absorption random walk on the TCL graph. The topic initiator will have the highest possibility to be visited. The second and third terms in Equation 24 represent the jumping possibility from a inlink node and a semantic relevant node, respectively.

Two special cases:

(1) $\alpha \neq 0, \beta = 0, \gamma = 0$. Equation 24 becomes $r_i = r_i^\dagger$. In this case, we just use the initial ranking score.

(2) $\alpha = 0, \beta \neq 0, \gamma \neq 0$. In this case, we ignore the initial ranking score and only consider the link and the time-related semantic relationship.

If we define the importance Ψ_i as the weighted combination of the inlink nodes size and the overall semantic relationship with following nodes, i.e.,

$$\Psi_i = \frac{\beta N_i^I + \gamma \sum_{j\in V_i^S} w_{ij}}{\beta + \gamma} \quad (25)$$

Equation 24 becomes a simpler version,

$$\begin{aligned} r_i &= \frac{\alpha}{\alpha + \beta + \gamma} r_i^\dagger \\ &+ \frac{\beta}{\alpha + \beta + \gamma} \sum_{<j,i>\in E_L} \frac{r_j}{N_j^O} \\ &+ \frac{\gamma}{\alpha + \beta + \gamma} \sum_{<j,i>\in E_S} \frac{w_{ij} r_j}{W_j^O} \end{aligned} \quad (26)$$

5. EXPERIMENTS

This section reports experiment results on real web data to demonstrate the effectiveness of our framework and the performance of our topic initiator ranking algorithms.

5.1 Data, Topics and Queries

Our data are all webpages from web search result. We investigate 332 topics related to three types of interest: product vegemite, cloud computing and smart grid. Overall 86,949 webpages are involved in our experiments. Table 2 shows some example topics.

Table 2: Example topics

Topic name
USA banned Vegemite
Google IBM cloud computing universities
IBM announces Blue Cloud
Google enters Smart Grid and announces PowerMeter
Xcel Energy announces first Smart Grid city

For each topic, to simulate the real web search situation, we try different queries with variant words and query length, because different users may submit different queries for the same topic. In total 916 queries are performed for those topics.

Ground Truth. We manually checked the web documents of each topic to identify the topic initiator. If there are multiple relevant web documents that appear in the earliest date and it is hard to identify a single topic initiator by the information available, we deem them as equally possible topic initiators.

Evaluation Measures The true topic initiator should be ranked as high as possible (ideal case is ranked as top 1). We evaluate the performance by rank r, the ranking order of the initiator in the ranking list, if there are multiple initiator candidates, choose the best rank. The overall performance is $Rank = \sum r_i/n$, the average rank of all queries (n is the number of queries). The standard deviation is reported to show the robustness of the algorithms, and we call this *Rank std* for short.

PageRank and HITS only work when the link information is available. If there is no link between any web documents of a query, all of them get the same score by the purely link-based algorithms. In such case, we use random guess and set the ranking of the topic initiator as $N/2$, i.e., half the number of the web documents.

5.2 Overall Performance

Figures 3 and 4 show the overall performance of the algorithms. ComboRank performs better than individual indicators, both in *Rank* and *Rank std*. InitRank achieves the best performance. Not only being able to dig out the true topic initiator and rank it in the very top, InitRank also has very small *Rank std* which indicates its performance is very robust among all the queries.

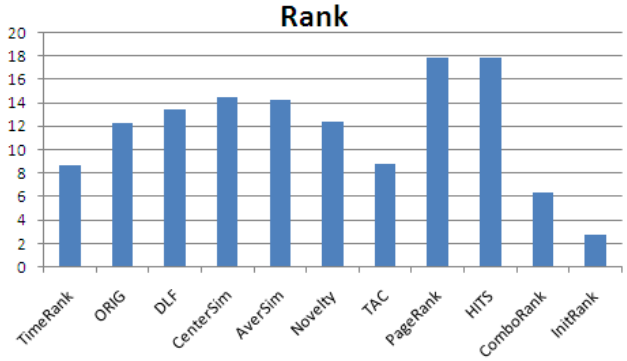

Figure 3: Overall performance of the algorithms. Y-axis denotes the *Rank*.

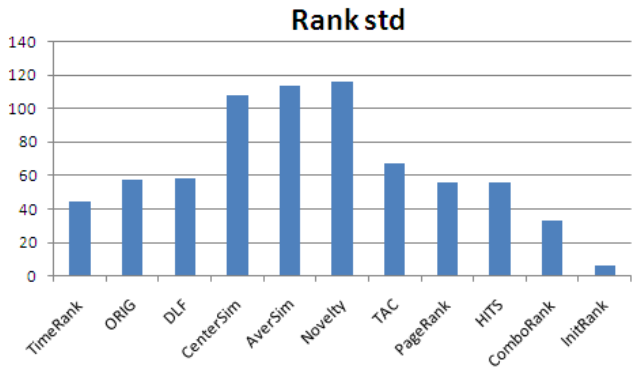

Figure 4: Standard deviation of Rank.

5.2.1 *Sensitivity of Parameter*

To simplify experiments, set $\alpha + \beta + \gamma = 1$ and $\beta = \gamma$ to reduce the original three parameters to only one, i.e., $s = 1-\alpha, s \in [0, 1]$. Figure 5 shows the sensitivity of InitRank on parameter s. When $s = 0$, InitRank only use initial ranking and no refinement based on the graph random walk is used, so the performance is not good. When $s = 1$, initial ranking score is totally ignored and thus dramatically degrades the performance. When $0 < s < 1$, initial ranking score and graph refinement are integrated to show good performance. The highest performance is achieved around $s \in [0.05, 0.2]$.

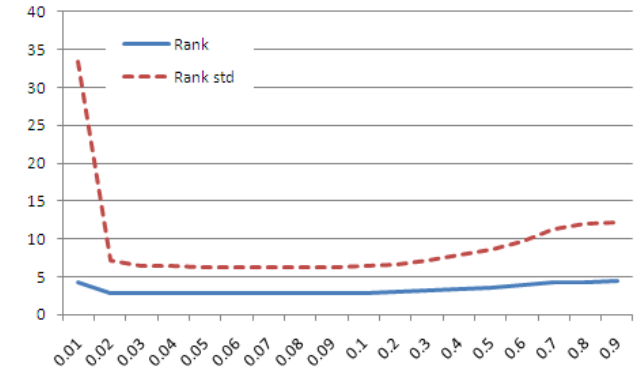

Figure 5: Sensitivity of parameters. X-axis denotes the value of s.

5.2.2 *Convergence*

Figure 6 shows the average ΔR and its standard deviation (std) over all the queries at each iteration. Because its value at iteration 1 is too big and dominates the figure, we draw the curve beginning at iteration 2. We can see from the result that InitRank converges very fast, only 5 iterations are enough for most cases.

5.3 Case Studies

We present two detailed case studies from two interesting topics: (1) "Vegemite ban" and (2) "Google IBM cloud computing".

Table 3 shows the statistics of the two topics. Figure 7 and 8 show the number of web documents per day for them.

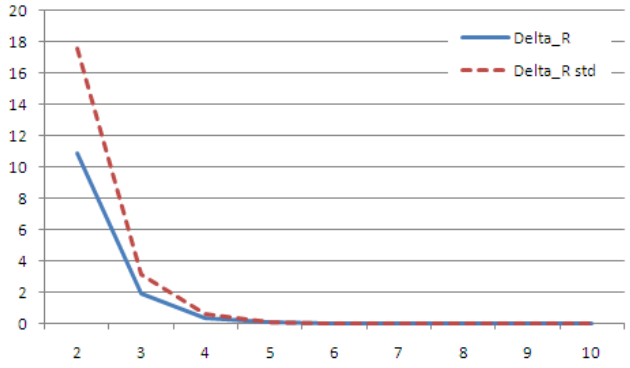

Figure 6: Convergence of InitRank. X-axis denotes the number of iteration. Y-axis denotes ΔR(i.e., $Delta\_R$) and its standard deviation.

Table 3: Statistics of the two case study topics

	Topic 1	Topic 2
# of web documents	4250	750
# of webpages	1497	729
# of websites	813	396
# of outlink websites	1093	948
# of outlink webpages	2942	3282

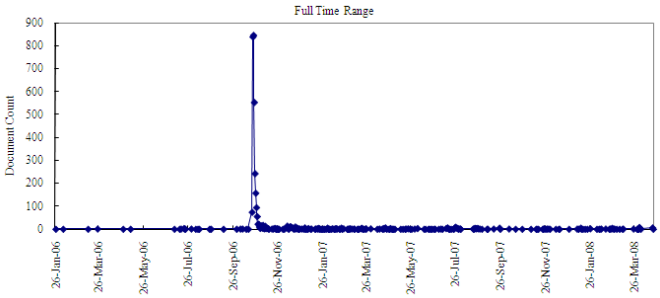

Figure 7: Number of web documents per day for "Vegemite Ban".

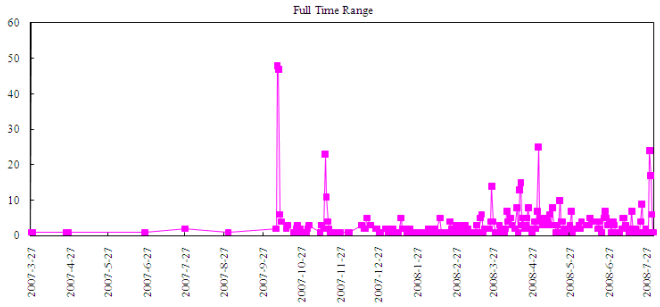

Figure 8: Number of web documents per day for "IBM Google Cloud Computing".

5.3.1 *Vegemite Ban*

Background: Vegemite is a food paste used mainly as

a spread on sandwiches and toast. The food is similar to British and New Zealand's Marmite and to Swiss' Cenovis. It is very popular in Australia and New Zealand. In October 2006, a rumor spread that Vegemite had been banned in the United States.

Facts: The fact behind the spreading of the topic is that: On October 5th, 2006, a blog article [17] first talked about Vegemite ban, and several people commented on the blog. However, it did not generate a burst. A very interesting thing is that the blog author of [17] is an Australian writer best known for his speculative fiction. This blog mentioned that "Around the same time, however, Deb Layne (Wheatland Press), my fine and lovely publisher, had a birthday, and I thought I can send her a jar of vegemite... Except of course, vegemite cannot legally be imported into the States anymore because it contains Folate." Wheatland Press, founded by Deborah Layne, is an independent book publisher specializing in science fiction and fantasy, and has published works written by the blog author.

On October 21, 2006, News Australia website published a news article reporting that USA banned Vegemite [2], and generated a burst on Vegemite ban. A lot of web documents have cited this article. On the next day, the same website published another news article titled "US bans Vegemite" [10], some web documents have cited this one.

On October 25, 2006, USA denied imposing ban on Vegemite [3]. Even though "USA bans vegemite" has been proven to be a rumor since then; there were still many lingering spreading of the rumor after that.

In the following we quote two articles which followed the topic and analyzed where it started from.

Quote [15]: It all started out with an article in Sunday's Courier Mail by Kelvin Healey, which was taken from Danny Lannen's article in the Geelong Advertiser. Online forums were on fire with the news of Vegemite being banned.

Quote [8]: Australians are particularly unhappy. (Kelvin Healey, "US bans Vegemite", The Courier Mail, Oct. 22; News.com.au, Oct. 21; Tim Blair via Dylan). If you're an American fan, act fast before eBay shuts down the auction.

The two examples show that even human may not be able to correctly identify the topic initiator. [15] falsely said it started with [10]. [8] listed several candidates (we surmise that the author did this partially because of not knowing which one is the true origin), but [17] was missing.

Results Table 4 shows the ranking result of the true topic initiator from different algorithms. If we just sort the web documents by time, blog [17] is only ranked 110th. All web documents appearing before it are not talking about Vegemite ban. Well-known link-based algorithms PageRank and HITS do not rank blog [17] as top 1 because there are some web documents which get much larger amount of citations. InitRank correctly ranks blog [17] as the 1st.

Table 5 shows the snippet for the top 1 result by InitRank and ranking only by time. We can see that the top 1 result of ranking by time just happens to have the query words "vegemite" and "ban", but does not really talk about the vegemite ban topic.

Interesting Finding: The interesting thing for the result is that we find that blog article [17] was actually the first to talk about Vegemite ban, while majority people through a standard analysis would conclude that the rumor started with the News Australia article [2] which appeared early and was cited by over 156 webpages.

Table 4: Ranking Result for "Vegemite Ban"

Algorithm	Rank of [17]
TimeRank	110
ORIG	657
Centrality	537
Novelty	305
PageRank	10
HITS	5
ComboRank	9
InitRank	1

Table 5: Top 1 result of the algorithms

Algorithm	Snippet
TimeRank	locate the *vegemite* and write my name on everyone ...I fought my way into parliament, and made a law *banning* Nuttelex [7]
InitRank	I just found out that *Vegemite* is *banned* in the States ...

5.4 IBM Google Cloud Computing

Background: In October 2007, IBM and Google officially announced to work together on cloud computing and collaborate with six USA universities.

Facts: On October 7, 2007, article [13] talked about IBM and Google would officially announce they will work together on cloud computing. On October 8, 2007, a lot of websites published this announcement, including the IBM official website [11] and many news websites.

Quote [13]: Google and International Business Machines are announcing Monday a major research initiative... The two companies are investing to build large data centers that students can tap into over the Internet to program and research remotely, which is called "cloud computing".

Results Table 6 shows the ranking result of different algorithms. [13] is correctly ranked as top 1st by InitRank.

Table 6: Ranking Result for "IBM Google Cloud Computing"

Algorithm	Rank of [13]
TimeRank	4
ORIG	84
Centrality	66
Novelity	21
PageRank	118
HITS	118
ComboRank	2
InitRank	1

Table 7 shows a quote of the top 1st result for InitRank and ranking only by time. The result also proves that ranking based on only time information is not enough.

Interesting Finding: The interesting thing for the result is that we find that the IBM-Google cloud computing collaboration is officially announced on October 8, 2007, but article [13] seems have published the news one day early.

6. CONCLUSIONS

In this paper, we introduce a new Web Mining and search

Table 7: Top 1 result for the algorithms

Algorithm	Snippet
TimeRank	...eBay, *Google* and Yahoo can be quite different than traditional IT shops.... more custom solutions than its competitors, including *IBM*..."*Cloud computing* is all about companies getting as many computers..."[14]
InitRank	*Google* and *IBM* join in '*cloud computing* research'...

technique/service - Topic Initiator Detection (TID) on the Web. When a user is interested in a topic, and wants to know which web document initiated the topic or was the first to talk about this. The service can answer this question.

We design a framework solution for TID on the Web and present InitRank, which is based on our proposed topic initiator indicators and refinement over a TCL graph, to rank the possibility of a web document to be the topic initiator.

Experiments are done with real web datasets, compared with approaches such as intuitive method of simple time sorting, well-known link-based algorithms PageRank and HITS, InitRank gets the best performance. InitRank can find the correct topic initiator in some cases where even human can make mistakes.

Our experiment results have interesting findings: one is that we found the first web document related to the Vegemite (a popular food product in Australia) ban, which can help the company conducts investigation; another is an article that published the IBM Google collaboration on Cloud Computing one day before the official announcement.

7. ACKNOWLEDGMENTS

We thank gratefully IBM Corporation for its support of this research. We thank Ying Chen, Bin He and Zhijun Yin for helpful discussions and suggestions.

8. REFERENCES

[1] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the 1998 DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[2] N. Australia. There's no accounting for taste. `http://www.news.com.au/story/0,23599,20623973-2,00.html`.

[3] N. Australia. Us denies imposing ban on aussie vegemite. `http://www.news.com.au/heraldsun/story/0,21985,20641682-663,00.html`.

[4] F. C. Lexical analysis and stoplists. In *Information Retrieval: Data Structures and Algorithms*, pages 102–130, Englewood Cliffs, New Jersey, 1992. Prentice Hall.

[5] S. Chakrabarti, B. Dom, P. Raghavan, and S. Rajagopalan. Automatic resource compilation by analyzing hyperlink structure and associated text. In *7th World Wide Web Conference (WWW'97)*, pages 65–74, 1997.

[6] I. De and A. Kontostathis. Experiments in first story detection. In *Proceedings of the 2005 National Conference on Undergraduate Research (NCUR)*, 2005.

[7] Filmfreke. `http://filmfreke.livejournal.com/187583.html` (2006-01-05 03:13:00) (Accessed 2008).

[8] T. Frank. Latest nanny state ban: Vegemite. `http://overlawyered.com/2006/10/latest-nanny-state-ban-vegemite/` (Accessed 2008).

[9] D. Hawking and P. Thistlewaite. Relevance weighting using distance between term occurrences. Technical Report Computer Science Technical Report TR-CS-96-08, Australian National University, 1996.

[10] K. Healey. Us bans vegemite. `http://www.news.com.au/couriermail/story/0,23739,20620744-953,00.html`.

[11] IBM. Google and ibm announce university initiative to address internet-scale computing challenges. `http://www-03.ibm.com/press/us/en/pressrelease/22414.wss` (Accessed 2008).

[12] R. Kosala and H. Blockeel. Web mining research: A survey. *ACM SIGKDD EXPLORATIONS*, 2000.

[13] S. Lohr. Google and ibm join in 'cloud computing research'. `http://www.iht.com/articles/2007/10/07/business/cloud.php` (Accessed 2008).

[14] D. Needle. Dell targets cloud computing. `http://www.internetnews.com/ent-news/article.php/3668201` (March 27, 2007) (Accessed 2008).

[15] Neil. Vegemite ban or cheap shot at the us? `http://melbourne.metblogs.com/2006/10/24/vegemite-ban-or-cheap-shot-at-the-us/` (Accessed 2008).

[16] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford University, 1999.

[17] B. Peek. Vegemite banned in the usa? `http://benpeek.livejournal.com/481233.html` (Accessed 2008).

[18] Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[19] M. Porter. Porterstemmer. `http://www.tartarus.org/~martin/PorterStemmer` (Accessed 2008).

[20] S. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of ACM SIGIR'94 Conference*, pages 232–241, 1994.

[21] A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the ACM SIGIR96 Conference*, pages 21–29, 1996.

[22] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *21st Annual international ACM SIGIR Conference on Research and Development in information Retrieval*, pages 28–36, New York, NY, 1998. ACM.

[23] Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. In *Eighth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, pages 688–693, New York, NY, 2002. ACM.