Supporting Analysis of Future-related Information in News Archives and the Web

Adam Jatowt, Kensuke Kanazawa, Satoshi Oyama and Katsumi Tanaka

Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto, Japan
Phone: +81-75-7535969
{adam,kanazawa,oyama,tanaka}@dl.kuis.kyoto-u.ac.jp

ABSTRACT

A lot of future-related information is available in news articles or Web pages. This information can however differ to large extent and may fluctuate over time. It is therefore difficult for users to manually compare and aggregate it, and to re-construct the most probable course of future events. In this paper we approach a problem of automatically generating summaries of future events related to queries using data obtained from news archive collections or from the Web. We propose two methods, explicit and implicit future-related information detection. The former is based on analyzing the context of future temporal expressions in documents, while the latter relies on detecting periodical patterns in historical document collections. We present a graph-based visualization of future-related information and demonstrate its usefulness through several examples.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms

Keywords

Future-related information retrieval, event prediction, temporal information analysis

1. INTRODUCTION

There is much written information available nowadays that is related to the near or distant future. As people want to organize and arrange their lives they publish documents containing plans, schedules, expectations or predictions on the future course of actions of objects or events. For example, companies like Apple may announce release of a new product or events such as Euro Soccer Cup may be planned.

People always wanted to know the future to increase their chances of achieving success and to alleviate the fear of unknown. Thus they often try to guess the future by directly extrapolating the past or by searching for any early indicators of future events. As the Web and large news corpora reflect society trends, behaviors and expectations they constitute powerful data sources for future event prediction. Correctly organizing and summarizing future-related information would help users to predict interesting events and their most probable dates (e.g. release date of a new model of a Toyota car, scheduled events in a given touristic place). It should also result in better planning and decision making (e.g., to buy or not a Toyota car now, to visit or not the given place).

Future-related information is inherently uncertain and variable when compared to the information on past events. Therefore users should analyze many different information sources in order to assess the credibility of future-related information. Another problem with future-related information is that it requires much time to be found, analyzed and summarized. One should also properly take into account the temporal aspects of the information. For example, the creation dates of documents or the time length of the predictions should be investigated when agglomerating the data. Due to the above issues it is not easy for users to manually find, analyze, compare and summarize future-related information using conventional search engines or state-of-the-art applications.

Although there have been some applications designed for mining and summarizing Web data such as QA or blog opinion mining systems, little research has been done so far on the analysis of future-related information on the Web or in other document collections. Considering the fact that manually agglomerating and organizing such information is rather difficult for users due to its amount and variability, we propose an automatic approach to support users in this task. The application that we demonstrate produces visual summaries of future-related information for a class of queries that are named entities such as names of persons, places, organizations, events.

We follow two directions to realize our objective. One is to effectively agglomerate explicit information that appears in the content of documents. This is done through the analysis of temporal expressions occurring in documents and the context in which they are used. The other one is to use the past data such as statistics on document creation over time as a basis for forming predictions of periodical events. We call these methods explicit and implicit future-related information detection, respectively.

JCDL'09, June 15–19, 2009, Austin, Texas, USA.

For both the methods we use search interface of the Google News Archive[1], which is a large online collection of current and historical news articles. In addition, we also adapt the explicit future-related information detection method to the Web, as it contains different kinds of documents. Thus, we use two types of documents. The first one is time-stamped documents such as news articles, which are also usually credible and up-to-date. Another type of documents is arbitrary Web pages that may not have any associated timestamps and whose trustworthiness levels can be quite different.

The remainder of this paper is structured as follows: In Section 2 we discuss the related work. In Section 3 we present our approach for detecting and summarizing future-related information. In Section 4 we show the results of experiments and demonstrate several examples. We also discuss the shortcomings of the proposed methods and our future work. Lastly, we conclude the paper in Section 5.

2. Related Work

To the best of our knowledge, there are no similar works except for the visionary paper by Baeza-Yates [1]. The author discussed the problem of "future retrieval" and the challenges that it poses. The paper also introduces a basic method for searching and indexing documents according to their future-related information. Each document would contain tuples of time segments and confidence probabilities about the occurrences of future events. The ranking score would be defined through the extension of traditional document ranking models by including timestamps and confidence probabilities. The system proposed in [1], in contrast to ours, does not analyze or aggregate information.

Kimura et al. [6] proposed mining the Web for re-constructing person histories and presenting them as a chronological table. Their method focuses on namesake disambiguation, detection and normalization of date expressions as well as extraction of personal information that accompanies past date expressions. Temporal expressions are extracted using pre-specified language patterns. In order to find information relevant to person's history, the authors analyze HTML tree structure and use machine learning methods. Their objective is to handle past information on persons, while our method is designed for analyzing and visualizing future data related to any named entities.

Question answering systems are to some extent related to our work. For example, Pasca [9] proposed a question answering system focusing on "when?" type questions. The application extracts information related to user answers by document surface processing using regular expressions. In addition, it indexes the answers according to extracted year dates.

Our second method for analyzing implicit future-related information is, to some extent, related to previous attempts for stock movement or sales volume prediction [11,3,4]. Wuthrich et al. [11] proposed the prediction method of stock indices using past news about companies and past stock indexes as a training data set. Their method predicts whether the stock indices in next day will be up, down or unchanged by extracting salient keywords from past news. Choudhury et al. [3] tried to predict changes in stock prices from blog communication patterns. By analyzing communication dynamics in the blogosphere, they managed to determine and visualize the most probable movement of stock prices using SVM. Gruhl et al. [4] showed that the volume of blog postings can be used to predict spikes in actual consumer purchase decisions on the example of books. The above works aim at prediction of the dynamics of stock prices or products' sales, while we detect and analyze future information about periodical events related to user queries.

Some researchers performed also sentiment analysis for predictions of product sales performance [8,7] or for analyzing emotional perception of the future [10]. Mishne and Glance [8] and Liu et al. [7] applied sentiment analysis methods to Weblog data for predicting movie success. Pepe and Bollen [10] analyzed the public mood concerned with the future on the basis of emails submitted to futureme.org, a Web service that allows scheduling emails to be sent at future dates.

3. FUTURE-RELATED INFORMATION ANALYSIS

In Figure 1, we show the overview of a system that we have built using our approach. It outputs visual summaries of future-related information about user queries using two methods. The first relies on collecting content containing explicit future information within analyzed documents. The second is based on calculating the probability of next instances of periodical events. For brevity, we will call them explicit and implicit methods, respectively.

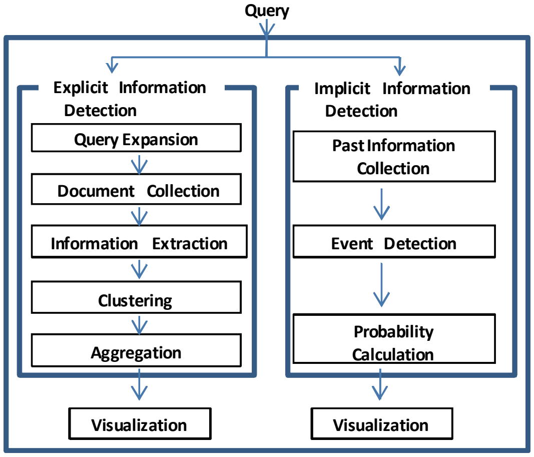

Figure 1: System image.

As basic underlying data we use snippets (i.e., short summaries) of news articles obtained from Google News Archive search engine. In addition, we also implemented the explicit method on the Web using Web page snippets obtained by the Yahoo! Search Engine API[2]. This implementation required adapting the method to the Web data, for which it is difficult to correctly detect document timestamps in contrast to news articles. Although some attempts were proposed for estimating the age of page content, they are rather costly and have varying precision [5]. Lastly, one needs to remember that the credibility and relevance of news articles are usually much higher than that of arbitrary Web pages.

[1] http://news.google.com/archivesearch

[2] http://developer.yahoo.com/search/

3.1 Explicit Method

3.1.1 Basic Concepts

When analyzing future-related information that is expressed in documents one needs to consider several issues. First the document creation time[3] (timestamp) needs to be taken into account, as it determines whether the document refers to the future, past or present. Consider an example shown in Figure 2. Doc1 was created in 2007, and the time to which the information in this document refers to is 2008. We call the latter the focus time of the document. As the current reading time is 2009, hence, Doc1 does not contain any future-related information at the reading time. On the other hand, the focus time of Doc2 is 2010, so it currently contains future-related information.

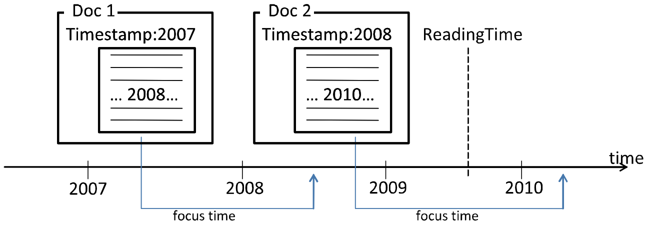

Figure 2: Visualization of three notions of time.

In general, in this work we distinguish three notions of time related to documents: document timestamp, focus time of document content and time of document reading[4]. Note that there can be also other notions of time included, such as document last-modification date, however, for simplicity, we neglect them here.

Second, temporal expressions in documents are roughly categorized into absolute and relative ones. The former explicitly identifies given time points or periods, for example, January 21st, 2009 or 2019. On the other hand, relative temporal expressions (e.g., 2 years later, next year) can only be anchored in time (i.e., normalized) when they refer to some absolute expression. Often, document timestamp serves as an anchoring point for normalizing relative temporal expressions occurring in documents. We use this simple approach in our method.

Third, temporal expressions can have different granularity levels: days, months, seasons, years, etc. For simplicity, we focus here on year dates and we leave investigating the finer granularity approach as our future work.

Lastly, any future-related information has its level of uncertainty that should be analyzed when constructing results. We follow here the majority rule, in which the probability of future event occurrence depends on the amount of sources in the collection that support this event. The time of the event is decided by the agreement between the future-related temporal expressions appearing in these documents. In other words, we compare the focus time of documents about the same events. In addition, we consider the timestamps of the documents when agglomerating the results. Here we assume that future-related information in more recent documents is generally more correct than the one in older documents. Thus if two documents mention the same future event but differ in details, such as the expected time of this event, then we put more trust into the younger document rather than into the older one. Some other solutions could be also used here such as investigating the character of documents (e.g., blogs vs. company homepages) or measuring the importance of documents in collections (e.g., by estimating the amount of readers/visitors of news sources or Web sites).

3.1.2 Collecting Documents

First a user inputs a query. The query is then sent to Google News Archive in order to collect relevant document snippets. Returned snippets contain the titles of news articles and their content parts that include the user query. Effectively, the snippets can be viewed as short query-based summaries of news articles created in order to indicate their relevance to the user query. For brevity, from now on, we will call document snippets as documents and the collection of document snippets as the collection of documents.

Sometimes data returned by the Google News Archive Search can be dominated by the same past event, especially when the event was widely reported by many news sources. Or, in another case, simply the most recent documents are returned. In both cases the results are biased and some important future-related information may be missing. For example, this can happen if the future-related information was reported before some bursty event concerned with the query or simply it appeared long time ago. In order to obtain fair sample of future-related information about the query we transform the query into a series of temporarily constrained queries. In other words we add temporal constraints to the query in the form of unit time periods. This requests the Google News Archive to deliver only the documents that were created within specified time periods. We construct several such queries constrained by consecutive and non-overlapping unit time periods within a certain time frame. In our implementation the time frame is from July, 2006 to December, 2008 and the unit time period has the length of six months. Thus, in total, there are five temporarily constrained queries. We issue these queries to the search interface of the Google News Archive and collect ten documents for each unit query. Figure 3 shows how the original query is divided into a series of temporarily constrained queries. Note that in the case of the Web, we obtain page snippets without this kind of query reformulation, as conventional Web search engine APIs usually do not allow issuing temporarily constrained queries.

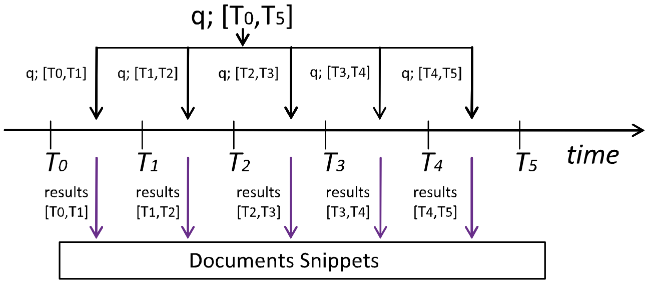

Figure 3: Query reformulation into a series of temporarily constrained queries.

The simple strategy of reformulating one query into a series of temporally constrained queries will typically return many documents. However, there may be still not enough data containing future-related information. To obtain sufficient number

[3] In case of Web pages the expression "content creation time" is more accurate as different content parts can be created at different time.

[4] We assume the time of document reading to be same as the time of issuing query.

of documents about the future, we also expand the query with some future dates e.g. "2012". In this way, we construct a set of expanded queries composed of the original query terms and future year dates ranging from 2009 to 2017. Each expanded query is also reformulated into a series of temporarily constrained queries as we have discussed above. The complete query extension approach is shown in Figure 4.

Finally, we remove duplicate documents and documents that do not contain the user-issued query neither any future-related date.

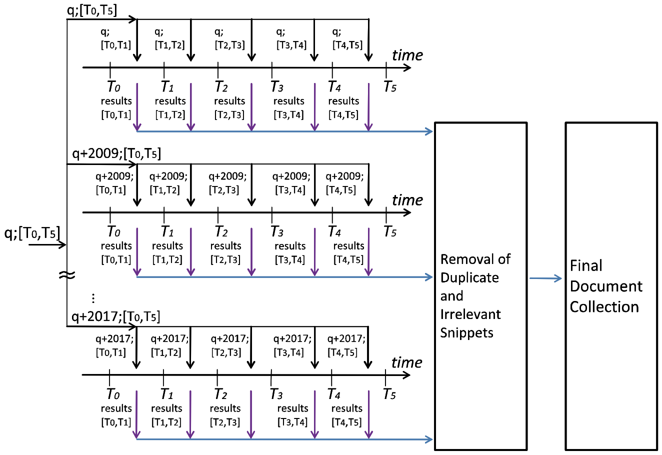

Figure 4: Data collection algorithm.

3.1.3 Focus Time Detection

In the next step we estimate the focus time of collected documents. This is done first by searching for temporal expressions inside the documents that refer to future year dates. We also use here a simple method for converting relative temporal expressions to absolute ones by treating document timestamps as anchoring points. For example, "10 years later" is converted to 2016 if the document timestamp is 2006. For Web pages we retrieve only absolute temporal expressions as detecting the age of page content is not trivial [5].

If a given document contains a single temporal expression which refers to the future, then this temporal expression is considered as the focus time of the document. In case there are several future-related temporal expressions, then the focus time is calculated as median of all future related expressions.

3.1.4 Clustering

Next, we cluster documents to find important future-related events. We use here k-Means, which is a partitional clustering method. We define the distance between documents based on the following observation. If two documents contain information about the same future-related event, then they should have similar content and the focus times of these documents should be also similar. Therefore the inter-document distance used in the clustering is defined by linearly combining the distances between document vectors and the distances between their focus times.

$$\begin{aligned} &Dist(doc_i, doc_j) = (1-\beta) * TermDist(doc_i, doc_j) \\ &+ \beta * TimeDist(doc_i, doc_j) \end{aligned} \tag{1}$$

β is a mixing parameter, *TimeDist(doci,docj)* is an absolute difference between the focus times of the documents, and *TermDist(doci,docj)* is the Euclidean distance between the feature vectors of the documents. Note that we do not consider here document timestamps as the documents created in different time periods could still refer to the same future events. Documents are represented using bag-of-terms approach with a prior elimination of stop words. As terms we use only nouns and verbs since they are often regarded to be the most informative entities. Feature vectors are then constructed using TF-IDF (term frequency – inverse document frequency) weighting scheme defined by the following expression:

$$tfidf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} * \log \frac{M}{m_i + 1} \tag{2}$$

$n_{i,j}$ is the occurrence frequency of a term i in a given document doc_j; m_i is the number of documents which contain term i and M is the number of all documents in the collection.

k-Means clustering needs a predefined number of clusters. Since it would be difficult for users to provide this number for arbitrary queries, we automatically estimate the optimum number of clusters using *Calinski-Harabasz*'s method [2]. This method measures the quality of clustering results trying to find a cluster combination in which the average distance between the documents within the same cluster (intra-cluster distance) is as small as possible and the average distance between the different clusters (inter-cluster distance) is as large as possible. Formally, it selects the number k ($k \geq 2$) of clusters by maximizing the following function:

$$CH(k) = \frac{\sum_{l=1}^{k} B_l / (k-1)}{\sum_{l=1}^{k} W_l / (M-k)} \tag{3}$$

B_l is the square sum of distances between cluster l and any other cluster and W_l is square sum of distances between the members of cluster l. Thus B_l represents the inter-cluster distance and W_l represents the intra-cluster distance. Both distances are measured using Equation 1.

After having finished the clustering we proceed to select the best clusters for constructing the summary of future events. First, we measure cluster quality by the following equation:

$$Q_l = \frac{B_l / (k-1)}{W_l / (N_l - 1)} \tag{4}$$

Here, N_l is the number of members in cluster l. According to the above equation, high quality clusters should describe different topics and also contain topically coherent documents. We select the top quality clusters using pre-fixed threshold level that is empirically estimated on large number of results.

3.1.5 Information Visualization

In order to visualize detected events the system makes cluster representations that characterize the main topic of each cluster. This is done by searching for documents whose feature vectors are closest to the centroid vectors of the clusters. Titles of such documents can be then shown to users as short descriptions of events represented by the clusters. Here we also consider the age of documents in order to find the most accurate description of events and their expected occurrence dates. The main assumption here is that the more recent a document is, the more probable is the future-related information that it contains. Thus more recently created documents are more suitable for cluster representations.

Documents used for representing clusters are selected using the following expression:

$$\text{Rep} = \left\{ \begin{array}{l} doc \in C \mid \\ \min\left[Dist(Centr, doc) * \exp\left(\dfrac{\text{Re}adTime - Date(doc)}{\lambda + 1} \right) \right] \end{array} \right\} \quad (5)$$

Date(doc) is document creation time, λ is a parameter which determines the influence of document's age (λ is set to 2 by default) and *Centr* is a centroid vector of a cluster *C*. The centroid vector is the vector containing means of term vectors of all the documents that belong to a given cluster.

Figure 5 shows the title of the document used for representation of a cluster that was obtained for the query "olympic summer". Under the title the top 5 terms of the cluster's centroid vector are shown for making it easier for users to understand the event represented by the cluster. The grey rectangles indicate the likelihood scores of the occurrence of the future event related to the cluster in the following years. In order to estimate the likelihood score in a given future year, we calculate the number of documents in the cluster whose focus time is equal to this year according to Equation 6.

$$R(t_i, C_j) = \sum_{\substack{doc_l \mid doc_l \in C_j \wedge \\ FocusTime(doc_l) = t_i}} \exp\left(\frac{\text{Re}adTime - Date(doc_l)}{\lambda + 1} \right) \quad (6)$$

Note that in the case of using Web pages as underlying data source we set *Date(doc$_l$)* to be equal to *ReadTime* in both Equations 5 and 6. This forces the exponent to be equal to 1.

In addition, we show also the information about the cluster size as the area of a red square. The larger the size of the square, the more widely discussed the event was. The red square is positioned in the year in which there is the highest likelihood of the event occurrence.

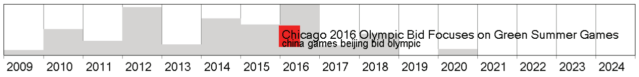

Figure 5: Visualization of a single event.

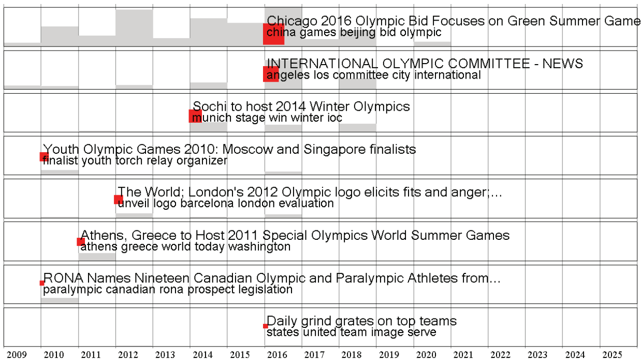

Figure 6: Complete visualization of the explicit future-related information for query "olympic summer".

In Figure 6, we show an example of the final visualization for the query "olympic summer". Representations of particular events are arranged vertically according to the sizes of their underlying clusters (the largest cluster at the top). When users want to learn more about a given event they can click on its representation to see the content of the whole snippet or to follow links to other related documents.

3.2 Implicit Method

In the previous subsection, we examined methods that extract future dates from document content. This is inadequate when future dates are more uncertain, conjectural or simply not available. We now describe our second, implicit, method that supports the identification of future events based on the extraction of patterns from past events such as sport events or new product releases. We re-construct the creation rate of related news articles over time and make predictions about future on the basis of detected events in the past. To this end, our method takes into consideration the bursts in the frequency of related news articles in the past. This approach can be used only for periodical events that have high probability to occur again in the future. Note also that the implicit method can be only applied on the collection of time-stamped documents; hence, we use here only the Google News Archive as an underlying data source.

3.2.1 Past Information Collection

User-provided query is used to collect information from the Google News Archive. An important challenge is the accurate reconstruction of the generation rate of related news articles over time. Figure 7 shows the overview of the method used for estimating document creation rate in the past. First, we divide the user query into a series of temporarily constrained queries in a similar way as in Section 3.1.2. This time however the main time frame spans 50 years and the granularity is set to 10 years, by default. Note that there is no need here for extending the query by adding future-related dates as in the explicit method, because we analyze the past now.

After sending the queries, the system takes the information on the number of the total results in the Google News Archive (i.e., hit count) for each query. Let *A[T$_i$,T$_{i+1}$]* denote the hit count for a unit time period *[T$_i$,T$_{i+1}$]*. This information however is still insufficient to correctly represent the rate of created articles in each year in the past as the granularity of temporal constraints was set to 10 years. Issuing queries with higher granularity (e.g., one year) would be however too time consuming (50 queries in total for 50 years' time span and one year's granularity). Instead, we estimate the rate of created articles in each year by checking the timestamps of a number, *N[T$_i$,T$_{i+1}$]*, of returned documents for each temporarily constrained query. The number of articles whose timestamps are to be analyzed in each time period is defined by the following equation:

$$N\,[T_i, T_{i+1}] = N * \frac{A\,[T_i, T_{i+1}]}{\sum_{k=0}^{n-1} A\,[T_k, T_{k+1}]} \quad (7)$$

N is the pre-fixed number of all articles to be analyzed. It is set to 500 by default. We retrieve timestamps of the top *N[T$_i$,T$_{i+1}$]* ranked documents for each temporary-constrained search query, *q;[T$_i$,T$_{i+1}$]*. Let *n$_q$[t$_i$]* denote the number of documents that have timestamp *t$_i$*, where *t$_i$* denotes a given year.

To increase the precision of this approach we should normalize *n$_q$[t$_i$]*. This is necessary because when the total time span is large there is high probability of obtaining skewed rate of documents in different time periods due to different journalistic activities or irregular news articles' crawling patterns over time. There are more articles in the collection that were created in the recent times than that were created long time ago simply because more news sources came into existence recently. In order to obtain unbiased results we thus should capture the pattern of overall journalistic activity in different time periods. We do this by issuing common words like stop words to the Google News Archive and measuring the average creation rate of documents containing them in the past by using the same approach as in Equation 7. The final article rate in a given year is now determined by the following equation.

$$n'_q[t_i] = \frac{n_q[t_i]}{avr_{a \in S}(n_a[t_i])} \qquad (8)$$

n'$_q$[t$_i$] is a normalized rate of documents in year t_i for a user query, while *n$_a$[t$_i$]* is the rate of news articles' creation in t_i obtained using a stop word a, and S is a set of stop words used for the normalization. S contains 20 the most common stop words. The average rate of documents obtained for the stop words need to be calculated only once and is stored within the system for later use.

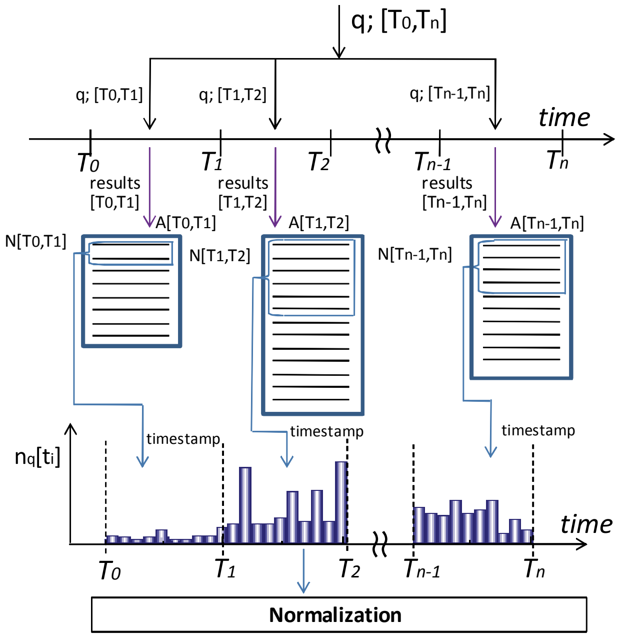

Figure 7: Overview of the method used for estimating the document creation rate over time.

3.2.2 Burst Detection

The next step is to determine important events related to the query. We do this by burst detection on the plot of the rate of news articles over time. We apply here a burst detection method that uses moving average. The time segments for which the rate of created news articles is higher than given cutoff values are regarded as the periods of the occurrence of important events. The cutoff value is determined by adding the value of moving average at a given year t_i and the average variance of news article creation rate over time.

$$cutoff\ (t_i) = MA_w(n'_q[t_i]) + \alpha * \text{var}(MA_w(n'_q[t])) \qquad (9)$$

MA_w is the moving average of a width w and α is a parameter determining the sensitivity-accuracy trade-off of the burst detection. If α is high then only few bursts can be detected, however, there is relatively high probability that these bursts will represent the true bursts of document creation rate over time. On the other hand, if α is low then many bursts can be detected, yet it is more probable that these are not "true" bursts representing past events (noisy bursts).

In Figure 8 we show an example of burst detection for query "olympic summer" that was issued to the Google News Archive for the time frame 1980 – 2008. Blue line represents the moving average with the window size set to three years. The colored bars denote time periods for which bursts were detected.

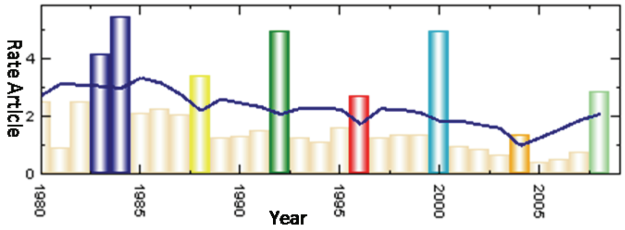

Figure 8: Burst detection for the query "olympic summer" over the time frame 1980-2008.

3.2.3 Periodicity

"Periodicity" describes how periodically an event occurs. It is used to calculate the probability of the future event occurrence. To calculate periodicity we use five parameters characterizing detected bursts: width, scale, center, span and content (see Figure 9). *Width(b$_k$)* means the length of the time period during which a given past burst b_k occurred. *Center(b$_k$)* denotes the center time point of the burst measured as the time at which the weight of the burst is largest. *Scale(b$_k$)* means how high is the burst at the time point indicated by *Center(b$_k$)*. Lastly, *Content(b$_k$)* is the centroid vector of TF-IDF-represented vectors of documents contained in the burst. The interval of the occurrence time between b_k and b_{k+1} is defined as:

$$Span(k, k+1) = Center(b_{k+1}) - Center(b_k) \qquad (10)$$

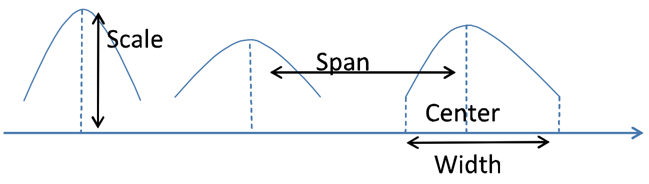

Figure 9: Parameters of a series of bursts of a given event.

Periodicity of an event E is then determined by five elements shown in Equation 11. The main idea behind the choice of these elements is that a periodical event should have bursts of similar shape and content that are also relatively uniformly spaced over time. For example, Summer Olympics should be reported in equal time distances with a similar strength and duration over time. The first element is the number of bursts: *elem$_{Num}$(E)*. It is assumed that the higher the number of bursts is, the higher the periodicity of the event is. Next, the standard deviation of interval between bursts: *elem$_{Span}$(E)* is used, since periodical events should have possibly same time span between the bursts. As the other elements we use the standard deviation of the width of bursts: *elem$_{Width}$(E)*, and the standard deviation of bursts' scale: *elem$_{Scale}$(E)*, both of which should be low for a highly periodical event. Finally, the average content distance between every pair of bursts: *elem$_{Content}$(E)* is measured. The content distance should be possibly low if bursts are actually occurrences of the same event over time.

$$
\begin{aligned}
elem_{Num}(E) &= \frac{1}{C_E} \\
elem_{Span}(E) &= \frac{\sqrt{\mathrm{var}_E(Span)}}{avr_E(Span)} \\
elem_{Width}(E) &= \frac{\sqrt{\mathrm{var}_E(Width)}}{avr_E(Width)} \\
elem_{Scale}(E) &= \frac{\sqrt{\mathrm{var}_E(Scale)}}{avr_E(Scale)} \\
elem_{Content}(E) &= \sqrt{\sum_{k \in E} B_k}
\end{aligned}
\tag{11}
$$

C_E means here the number of bursts constituting an event E. $elem_{Span}$, $elem_{Width}$ and $elem_{Scale}$ are normalized by their respective average values. B_k is the square sum of content distances between bursts.

The more periodical the events are, the smaller should be the sum of above elements. The periodicity is defined as:

$$
\begin{aligned}
Per(E) &= e^{-\sum_{j \in S} w_j * elem_j(E)} \\
S &= \{Num, Span, Width, Scale, Content\}
\end{aligned}
\tag{12}
$$

where w_j is a weight assigned to each element. We assume that $w_{Num} \geq w_{Content} \geq w_{Span} \geq w_{Width} \geq w_{Scale}$ (by default 5,5,5,1 and 0.2, respectively).

3.2.4 Estimation of Event Probability

In the preceding section we defined periodicity to measure how periodical events are. In this section we describe the way to calculate the probability of the next occurrences of periodical events.

An important problem here is that some of detected bursts could represent noise and have little in common with an actual periodical event. Such noisy bursts would deteriorate the prediction. In order to detect noisy bursts we use an exclusion algorithm which eliminates certain bursts and re-calculates event's periodicity using only the remaining bursts. In this way, periodicity is re-computed for every possible combination of bursts and the combinations resulting in the highest periodicity can be then found.

Next, we predict the probability of the forth-coming occurrence of the event. Probability of the next occurrence of an event at time point t_i is estimated as the sum of probabilities of this event's occurrence resulting from all possible combinations of bursts. For judging the impact of each possible combination of bursts we use the value of its periodicity score. The most periodical combinations of bursts will influence the prediction most highly. For each burst combination we estimate the most probable time point of the new event occurrence using the average span calculated for this burst combination (see Figure 10). In other words, the time distance between the new event occurrence and the last burst of a given combination of bursts is equal to the average span value for this burst combination. The probability is expressed formally by Equation 13. Note that the score is normalized to represent the actual probabilities in future years.

$$
\begin{aligned}
\mathrm{Prob}(t_i) &= \sum_{E_j | E_j \subseteq E \wedge t_i = Center(New(E_j))} Per(E_j) \\
Center(New(E_j)) &= avr_{E_j}(Span) + Center(Last(E_j))
\end{aligned}
\tag{13}
$$

E_j is a given combination of bursts obtained by removing some bursts from the original series of bursts of E. $New(E_j)$ denotes a new event occurrence predicted from E_j, while $Last(E_j)$ is the latest burst in E_j.

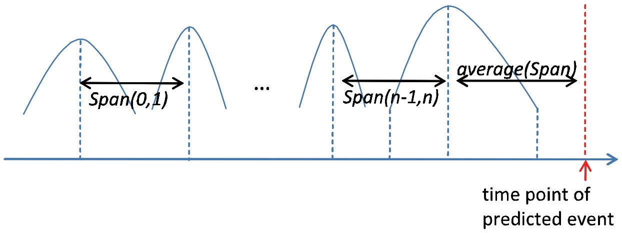

Figure 10: Prediction of future event occurrence on a given combination of detected bursts.

In Figure 11, we show final results obtained for query "olympic summer". Colored bars in the left-hand part denote detected bursts in the past, while the bars in the right-hand side denote the probabilities of the next event occurrence in future years. The bar in dark color in the right-hand side part indicates the outstanding probability. The probabilities are considered to be outstanding if they are 2 standard deviations higher than the average probability level. Note that we also show historical data of document creation rate for users to better judge the calculated prediction results.

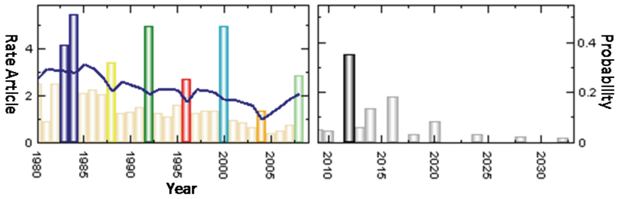

Figure 11: Probabilities of the next event occurrence for query "olympic summer" calculated on [1980-2008].

4. EXPERIMENTS

In this section we show experimental results of our methods. It is, however, difficult to evaluate the proposed methods as there is no standard comparison benchmark, neither other similar system available. We do a simple evaluation by calculating the average precision of the output generated by our methods and by discussing several example results.

4.1 Explicit Method

To calculate the precision of the explicit method we issue 20 test queries listed in Table 1. Then we manually analyze the returned clusters and the likelihood scores of expected events. We consider the clusters as good ones if the following three conditions are met. First, (a) we can know the main topic of an event from the description of its underlying cluster. Second, (b) the cluster indicates an important event related to the query. Third, (c) the actual expected time of this event is the same as the most probable occurrence time calculated by our method.

The left-hand side graph in Figure 12 shows the average precision for both news articles and the Web in relation to parameter β. The results are better for news articles than for the Web data. This is probably due to the higher credibility and accuracy of the former source as well as the feasibility of detecting news articles' timestamps. The precision is better for lower values of parameter β and achieves the highest value for $\beta=0.2$.

The right-hand side graph was obtained by considering only the conditions (a) and (b) in the calculation of the average precision. That is, the prediction of the time point of event occurrence has not been taken into account in this case.

Table 1: Queries used in the calculation of average precision for the explicit method.

barack obama	boeing new plane	chicago	dell	democracy
house bubble	inflation	invention	iraq	london
michael jackson	nato	new york	olympics	poland
soccer	toyota	war	windows	world cup

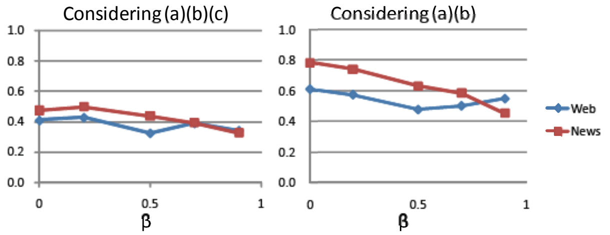

Figure 12: Average precision of the explicit method with (left) and without (right) consideration of prediction of event's time.

Below we discuss several example results. In all the examples we have set the parameter *β* to 0.7 for which the average precision should be lower than 60% according to Figure 12. Figure 6 in Section 3.1.5 shows the results obtained for query "olympic summer" sent to the Google News Archive. The largest two clusters shown at the top of the figure represent information about Chicago and Los Angeles being candidates for Summer Olympics in 2016. There is also information about Youth Olympic Games in 2010 with Moscow and Singapore being the finalists of the candidate selection process (4th cluster). The 5th cluster from the top is about London Olympic Games to be held in 2012, while the next cluster is on Special Olympics World Summer Games which will be held in 2011 in Athens. The remaining clusters including the one on Winter Olympics 2014 in Sochi have lower relevance to the query.

The next example that we show was obtained for query "poland" (Figure 13). The second top event is about UEFA European Football Championship (Euro 2012) that is going to be held in Poland and Ukraine in 2012. This is the most important event awaiting Poland in the near future as it is expected to considerably boost the local economy and test the country's abilities to organize a large international event. As there is much excitement these days about Euro 2012 in Poland, hence, the coach of the national soccer team, Leo Beenhakker is often put in the spotlight, especially as he is the first foreign coach of the Polish national soccer team in the history. The smallest size cluster is about Beenhakker's future as a coach of the Polish national soccer team and contains the information on the extension of his contract.

Another main event in the near future is the adaption of Euro currency instead of a current currency called "zloty". This is expected to take place around 2012 following joining the European Union (EU) by Poland in 2005. However, a referendum will be probably necessary as planned by Polish Prime Minister Tusk in order to convince the citizens. Top 3rd cluster is about this event. The top 1st cluster seems to be also partially related to Euro currency (terms: bond, EU, zloty), however it also contains much noise which seems to have decreased the quality of its representation. The remaining two clusters in this example are about current and forth-coming financial and fiscal problems of Poland.

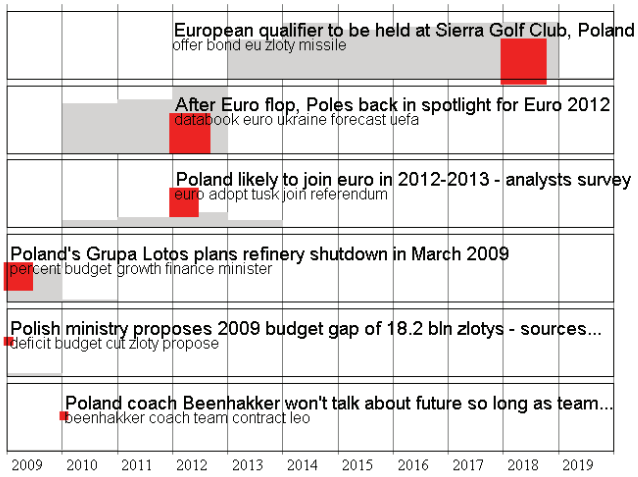

Figure 13: Results of the explicit method for query "poland".

Figure 14 demonstrates the results for query "Toyota" which is one of the largest international automakers. There are several events shown related to starting the sales of new cars or building plants for the production of these cars in the future: Venza in 2009 (4th cluster) and Plug-In Hybrid in 2010 in China (9th cluster). Other clusters are about predictions of Toyota's financial condition (5th cluster and 6th cluster), predictions of new-generation car (3rd cluster and 8th cluster) and comparison of Toyota with another automaker GM (2nd cluster and 7th cluster). The representation of the largest cluster is about a car crash hence this cluster is rather irrelevant.

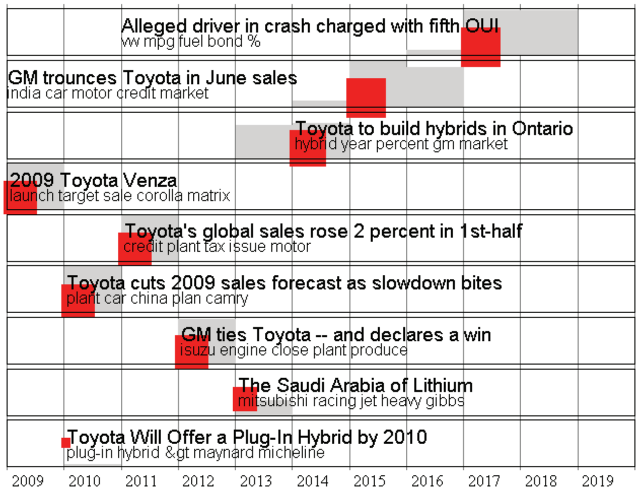

Figure 14: Results of the explicit method for query "Toyota".

Figure 15 is a visualization of future-related information for query "oil crisis". The first cluster is about the prediction of oil prices. The second one contains information about financial condition of a major oil company Chevron and about its Hebron heavy oil project at the east coast of Canada. Other clusters contain abstract topics related to oil crisis (3^{rd}, 6^{th} and 7^{th} cluster). Although, we cannot get clusters about exact predictions of oil crisis, the results contain information about the plans of major oil producers and future oil costs. This could help in further exploration of this topic.

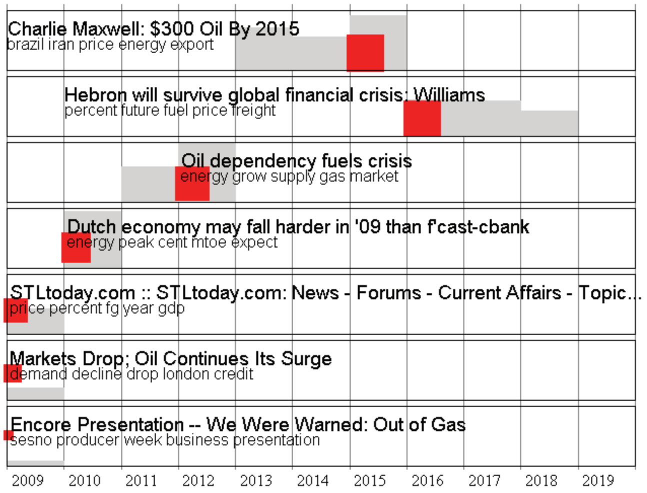

Figure 15: Results of the explicit method for query "oil crisis".

4.2 Implicit Method

In the evaluation of the implicit method we have checked the precision of burst detection and the precision of future event prediction. We have done the test on 10 queries shown in Table 2. Precision of the detection of the time of new event occurrence has been measured by checking whether the outstanding probabilities indicate the expected time points of the next event instances. We could find such expected time points only for five queries and the average precision was 60% (3/5).

Figure 16 shows the average precision and recall of the burst detection in relation to the values of parameters α and w. A given burst is considered to be a correct one if it indicates the actual event that occurred in the past. We can see that the precision does not change much for different values of α and w. On the other hand, as expected, the recall decreases for higher window sizes, w, and for higher values of parameter, α.

Table 2: Queries used in the calculation of average precision and recall of burst detection in the implicit method.

batman movie	el nino	financial crisis	nintendo console	oil crisis
olympic summer	presidential election	star wars	windows new release	world cup

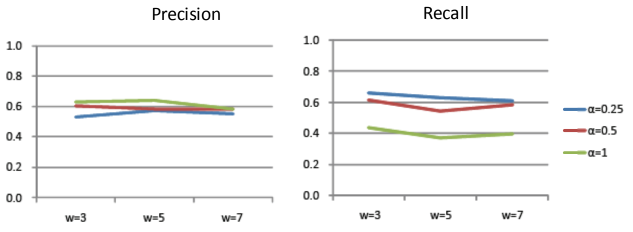

Figure 16: Precision and recall of burst detection for the implicit method depending on the values of *α* and *w*.

Below we discuss some examples of results returned by the implicit method. In Figure 11 the system correctly predicted the next Summer Olympics to be held in 2012. The bursts were correctly identified for the period from 1988 to 2008. However, the burst lasting from 1983 to 1984 was wrongly detected as the event was held only in 1984. It is because many news articles were written before the occurrence of the event.

In the example shown in Figure 17 we show the rate of document creation over time and the probabilities of the next event occurrence calculated for query "world cup" on the time period [1980-2008]. Every four years bursts were detected and the probability of the World Cup occurring in 2010 is the highest, which is a correct prediction. However, again there was a false estimation of the event width in two cases.

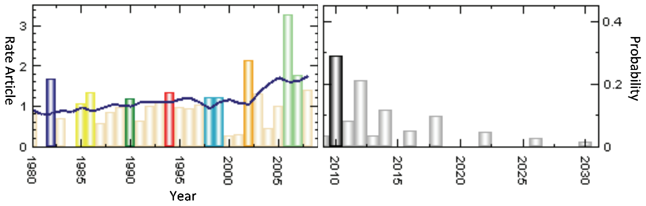

Figure 17: Past events and probabilities of the next event occurrences for query "world cup".

As another example we issued query "presidential election" on the time period [1970-2008] (Figure 18). Although, some past presidential elections were not detected (1980, 1984, 1988, 1992, 1996,) and some were wrongly detected (1971, 1974, 1978-1979), we could still provide correct results for the next presidential election in USA in 2012. The somewhat high ambiguity of this query was the probable reason for the poor event detection in the past.

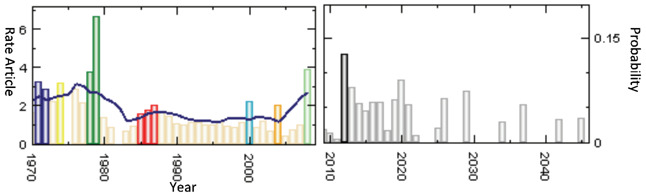

Figure 18: Past events and probabilities of the next event occurrences for query "presidential election".

Lastly, we show the results for query "oil crisis" on the time period [1970-2008] in Figure 19. We managed to detect 6 bursts in the past data, and predict two future events based on the selection of outstanding probabilities in 2010 and 2016. Currently, it is difficult to tell how probable this prediction is.

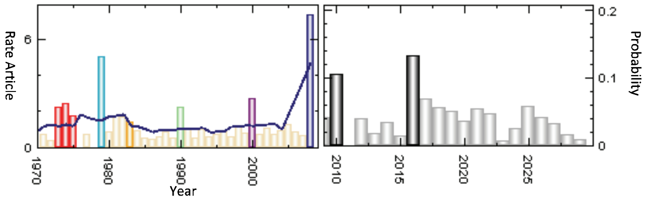

Figure 19: Past events and probabilities of the next event occurrences for query "oil crisis"

4.3 Discussion and Future Work

This research aims at providing supporting application for users to organize future plans, predictions and expectations regarding real-world objects. By mining large textual datasets we attempt to automatically construct the views of the most probable course of actions related to user queries. The applications like the one described above should help users to better coordinate their planned actions as they can better understand the current state of predictions or at least should have convenient starting points for further explorations.

There are several directions for improving the proposed method. First, both the explicit and implicit methods could be combined to produce single results. This could be for example realized by automatically selecting best input terms (e.g. terms characterizing some periodical events) for the implicit method based on the results obtained using the explicit method.

Second, extraction and normalization of temporal expressions should be improved. In this work we only consider yearly dates. Taking into account temporal expressions of the finer granularity like seasons, months or days should improve the precision. For example, in the explicit method considering shorter time units would enable detection of many annual events such as conferences, festivals or national holidays. In this work, we also do not capture the durations of time periods in temporal expressions such as "from 2010 to 2015" or "by 2012". Our method should be then extended to capture different linguistic patterns associated with temporal information in text. Other kinds of examined linguistic patterns could relate to the probability of future-related information. For example, expression "X is scheduled to happen in 2011" is more likely to denote an actually forth-coming event than the expression "X may happen in 2011". In addition, the date normalization scheme that we use could be further improved.

Third, we should also focus more on the credibility of underlying data sources in order to assign weights to documents depending on their trustworthiness levels. This could be realized through the examination of sources' characteristics such as their popularity, age, impact, etc.

Lastly, further extensions of the above methods can involve creating applications for continuous monitoring of content additions/deletions in underlying datasets, providing more complex user interaction mechanisms or allowing higher-level combinations of the results obtained for multiple, related queries.

5. CONCLUSIONS

Many times users want to know the expected future events related to given entities. Knowing the most probable future course of events should give them advantage to appropriately plan their actions. In this paper we propose two methods for supporting users in the process of future event analysis for their queries by using search interface of an online news archive collection and a Web search engine. The first method is based on detecting and summarizing future-related information in documents, while the second uses the re-constructed generation rate of news articles over time for the prediction of periodical events. The technology that we propose is a first step into building systems for automatic creation of reports on future events related to user queries.

6. ACKNOWLEDGEMENTS

This research was supported in part by the National Institute of Information and Communications Technology, Japan, by Grants-in-Aid for Scientific Research (Nos. 18049041, 18700111 and 19700091) from MEXT of Japan, by the Kyoto University Global COE Program: Informatics Education and Research Center for Knowledge-Circulating Society and by a Microsoft IJARC CORE4 project entitled "Toward Spatio-Temporal Object Search from the Web".

7. REFERENCES

[1] R. Baeza-Yates. Searching the Future. *Proceedings of ACM SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval (MF/IR 2005)*, 2005.

[2] T. Calinski and J. Harabasz. A Dendrite Method for Cluster Analysis. *Communications in Statistics*, vol. 3, no.1, pp.1-27, 1974.

[3] M. D. Choudhury, H. Sundaram, A. John and D. D. Seligmann. Can Blog Communication Dynamics be Correlated with Stock Market Activity? *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia*, pp. 55-60, 2008.

[4] D. Gruhl, R. V. Guha, R. Kumar, J. Novak, A. Tomkins. The Predictive Power of Online Chatter. The 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 78-87, 2005.

[5] A. Jatowt, Y. Kawai, and K. Tanaka. Detecting Age of Page Content. *Proceedings of 8th International Workshop on Web Information and Data Management*, pp. 137-144, 2007.

[6] R. Kimura, S. Oyama, H. Toda and K. Tanaka. Creating Personal Histories from the Web using Namesake Disambiguation and Event Extraction. *Proceedings of the 7th International Conference on Web Engineering*, pp. 400-414, 2007.

[7] Y. Liu, X. Huang, A. An and X. Yu. ARSA: a Sentiment-aware Model for Predicting Sales Performance Using Blogs. *Proceedings of the 30th Annual International ACM SIGIR Conference*, pp. 607-614, 2007.

[8] G. Mishne and N. Glance. Predicting Movie Sales from Blogger Sentiment. *Proceedings of the Spring Symposia on Computational Approaches to Analyzing Weblogs*, 2006.

[9] M. Pasca. Lightweight Web-Based Fact Repositories for Textual Question Answering. *Proceedings of the 16th ACM CIKM Conference*, pp.87-96, 2007.

[10] A. Pepe and J. Bollen. Between Conjecture and Memento: Shaping a Collective Emotional Perception of the Future. *Proceedings of the AAAI 2008 Spring Symposium on Emotion, Personality and Social Behavior*, 2008.

[11] B. Wuthrich, D. Permunetilleke, S. Leung, V. Cho, J. Zhang and W. Lam. Daily Prediction of Major Stock Indices from textual WWW Data. *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pp.364-368, 1998.