Information Retrieval with Time Series Query

Hyun Duk Kim, Danila Nikitin,
ChengXiang Zhai
Dept. of Computer Science
University of Illinois at Urbana-Champaign
{hkim277, nikitin2, czhai}@illinois.edu

Malu Castellanos, Meichun Hsu
Information Analytics Lab
HP Laboratories
{malu.castellanos,
meichun.hsu}@hp.com

ABSTRACT

We study a novel information retrieval problem, where the query is a time series for a given time period, and the retrieval task is to find relevant documents in a text collection of the same time period, which contain topics that are correlated with the query time series. This retrieval problem arises in many text mining applications where there is a need to analyze text data in order to discover potentially causal topics. To solve this problem, we propose and study multiple retrieval algorithms that use the general idea of ranking text documents based on how well their terms are correlated with the query time series. Experiment results show that the proposed retrieval algorithm can effectively help users find documents that are relevant to the time series queries, which can help users analyze the variation patterns of the time series.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*text analysis*

Keywords

Information Retrieval, Time Series Query, Text Stream

1. INTRODUCTION

In most existing work on information retrieval (IR), the problem involves a keyword text query entered by a user, and the goal is to retrieve documents from a collection of text documents, which are relevant to the query. While satisfying a user's information needs via keyword queries has so far been the primary application of information retrieval, in this paper we show that there is also another type of applications, where the goal is to find the documents in a text collection that can explain the changes of an external time series variables.

Topics discussed in text documents often have relations with other non-textual variables. For example, the reporting of a negative event about a company in the news might affect its stock prices. Similarly, the dramatic changes of the stock price of a company might also trigger discussions about it in the news. Another example is a new election pledge in a campaign that may affect support rate to the candidates in a presidential election.

In such cases, there is often a need to understand which topics in the text data may be correlated with the time series data. For example, investors are probably interested in what leads to the increase or decrease of the stock prices and use such relationships to forecast future price changes. Companies may want to understand how product sales rise and fall in response to such text data as advertisements or product reviews. In election campaigns, analysis of news or social media may explain why a candidate's support has risen or dropped in the polls. Such understanding of the situation can help to make better future strategies for sales or polls.

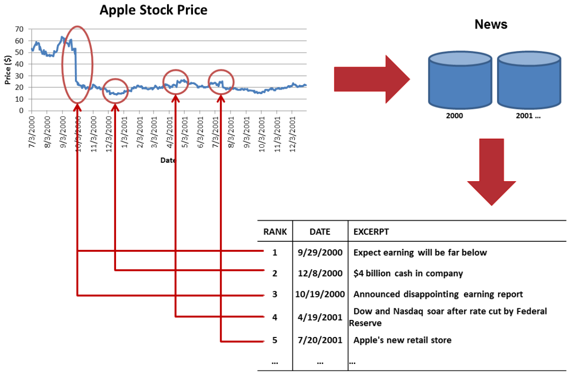
One way to help users find such relevant topics is to use time series as a query. Ideally, the documents retrieved from a time-stamped text collection will include those topics that are correlated with the given time series. For example, using Apple stock price time series query, we may be able to retrieve relevant documents from general news stream (Figure 1). The highly ranked documents are not only articles 'about' Apple company, but also potentially explain major changes of the input time series.

This is an interesting and novel retrieval problem, which has not been studied before. Although there were some previous works about text and time-series retrieval as we explain more in Sec 2, the main difference between this problem and other existing formulations is that instead of using a text query, we use a non-textual time series query. To solve this retrieval problem, the main challenge is how to cross the barrier between a time series query and text documents. Our key idea for solving this challenge is to first find terms whose frequency distribution over the time period of a collection is correlated with the query time series, and then use these terms to further find possibly relevant documents. We propose and study multiple retrieval algorithms to implement this idea. In general, we would first compute the word frequency curves from the text stream and calculate their correlations with the input time series query. Then, we would score a document based on the weighted average of the words in the document, where the weight is a function of the value of the correlation between the word and the input query.

We evaluate the proposed algorithms on the news data with stock prices. In addition to quantitative evaluation with standard information retrieval measures such as mean average precision (MAP) and normalized discounted cumulative gain (NDCG), we also qualitatively evaluate the top ranked documents to see if they are useful for understanding the changes in the input time series. Experimental results show that the proposed retrieval algorithms can effectively help users find documents that are relevant to the time series queries.

ICTIR '13, September 29 - October 02 2013, Copenhagen, Denmark.

ACM 9978-1-4503-2107-5/13/09 ...$15.00.



RANK	DATE	EXCERPT
1	9/29/2000	Expect earning will be far below
2	12/8/2000	$4 billion cash in company
3	10/19/2000	Announced disappointing earning report
4	4/19/2001	Dow and Nasdaq soar after rate cut by Federal Reserve
5	7/20/2001	Apple's new retail store
...	...	...

Figure 1: Using Apple stock price as a query to retrieve news articles.

2. RELATED WORK

The main goal of traditional information retrieval is finding relevant documents to the textual keyword query [33]. However, so far there have been no attempts to retrieve relevant documents directly from time series queries, which we study in this paper.

While there were studies in the time series matching and retrieval, they mainly focused on the retrieval from a collection of time series, and not from text data [1, 2, 3, 4, 6, 8, 14, 15, 16, 24, 26, 27, 32]. The main technical challenges for this problem are about the accurate measurement of similarity between two time series, and their efficient search. There are various methods for measuring similarity between time series, from basic measures such as Euclidean distance and correlation to more advanced techniques such as shape definition language [3], pattern matching and analysis [2, 6], longest common sequence matching [14], and dynamic time warping [32]. While these techniques are solely focused on numerical time series data without the consideration of related text data, they are useful tools for an approach to measuring similarity between the time series of word frequency and the input time series.

For more efficient and fast retrieval, various indexing and dimension reduction techniques can also be used, such as discrete Fourier transformation [1, 26, 27], wavelet transformation [4], and a probabilistic approach [15]. Like the general information retrieval problems, feedback on time series is also studied[16]. In this paper, we focus more on how accurately we can retrieve relevant and causal documents. These techniques can be possibly applied to speed up our method.

In addition to the previous methods, there were attempts to co-analyze text data with time series which showed interesting results. NTCIR proposed GeoTime [1] that is a geographical and time-dependent event retrieval task by question queries including location and time constraints. People have tried to summarize time series by extracting textual expressions [12, 31] and use their linguistic features to retrieve similar time series [30]. Temporal similarity analysis of word dynamics has helped to find semantically related terms [25]. Recent works showed that using different term weighting based on word dynamics over time, can help to improve document retrieval [5, 11, 19, 20]. Our retrieval problem shares a similar concept to these works in that we give different weights to terms, based on the time series analysis. However, we relate word dynamics to the input external time series which has not been considered in the previous works. Moreover, in problem setup, we use time series itself as a query, while previous works are still based on the keyword query retrieval task.

Research on stock prediction using financial news content is also relevant to our work [13, 23, 28]. One of its goals is to find the most predictive words and label the news according to their effect on the stock prices for that specific day. Most of these works are based on a regression or a classification problem setup, unlike our problem.

Causal topic mining [18] is a yet another closely related work. It automatically models topics that have causal relationships to the external time series input. However, in this paper, instead of topic retrieval based on probabilistic topic modeling, we try to retrieve relevant documents with word-level-based analysis. This approach is more simple and efficient, because we do not need complex parameter estimation steps of a probabilistic topic model.

3. RETRIEVAL WITH TIME SERIES QUERY

3.1 Motivation

Information retrieval with a time series query is a practical problem that arises in many text mining applications where there is a need to analyze text data and discover potentially causal topics in order to explain the changes of external time series variables.

Given an input time series and the knowledge about its topic, it would be natural to think that we could use the topic keywords with traditional IR techniques to obtain relevant documents that could explain the evolution of the external time series. However, there are three main advantages of a time series-based information retrieval approach over the traditional keyword-based information retrieval.

First, for the envisioned analysis problem, we need a special notion of relevance which goes beyond the traditional topical relevance. Specifically, we want to find articles containing topics that are highly correlated to the changes in the input time series. Traditional information retrieval does not consider *dynamics of terms* and how they are related to the external time series. Instead, it focuses on matching of query terms. In contrast, our technique measures the correlation between the time series of each term and the input time series query, and uses correlation to weigh the terms. In this way, we can rank documents based on how correlated their content is with the time series.

Second, the usage of time series as a query has an advantage of avoiding user bias in keyword selection that may give us misleading results, and irrelevant documents. For example, let us assume that Apple's iPhone sales revenue is our time series query. An analyst may use the company and product name as a query to find relevant documents that explain the behavior of the revenue. However, a competitor's product, such as Samsung's Android phone, may also be a good keyword if it had a significant effect on the sales of iPhone. We could add "Samsung" and "Android" as additional keywords, but still there could be even more related terms, corresponding to related entities, facts, and aspects that could have serious impact on the sales. For example, there could have been a new trend in the smartphone market or even a global economic trend, which the analyst would likely have missed. Therefore, rushing to specify the subtopic and using its keywords can lead to the loss of important hidden signals, related to the time series. By using the time series query itself, we let the algorithm find the most related documents corresponding to different signals that affect the behavior of the external time series.

Third, in addition to the previous examples of understanding stock prices, sales revenues, and election polls, we can even apply the technique to an unknown or an artificial time series curve. For example, we may have unknown traffic signals from the web and may be able to find potential reasons for the traffic by searching on social media sites such as Facebook and Twitter. For another

[1] http://metadata.berkeley.edu/NTCIR-GeoTime

example, we could analyze signals obtained from physical sensors. In that case, we might be able to find unknown reasons of the signal changes or glitches from the log data or other kinds of possibly related text articles (e.g. local news, and web articles of the locations where sensors are installed). Furthermore, in addition to the given time series as an input, users may also use multiple time series data to derive a normalized discriminative time series as a query, or even draw an artificial time series curve based on their needs to retrieve correlated documents with such an imagined time series.

3.2 Problem Definition

The input of our retrieval problem consists of a text stream and a time series query. The output is a ranked list of input text documents. Formally, given a numeric time series query with time stamps $TS = \{(x_1, t_1), (x_2, t_2), ..., (x_n, t_n)\}$, and a text stream which is a collection of documents with time stamps within the same time period, D = {(d_1,t_{d_1}), ..., (d_m, t_{d_m})}, we have to compute a ranked list of documents, D' = {(d'_1,$t_{d'_1}$), ..., (d'_m, $t_{d'_m}$)}. These are the documents that explain the behavior of the time series.

The main challenge in solving our problem is to cross the barrier between the query and the document. In regular text retrieval, both the query and the document are text objects, making it easy to match them, but in our case, the query is a non-textual time series. We describe our methods to solve this challenge in the next section.

4. METHOD

4.1 General Method

The goal of our retrieval problem is to find documents having contents that could possibly help to explain the changes in the input time series. In order to cross the barrier between a non-textual query and a text document, our key idea is to first obtain words whose frequency distributions over time are correlated to the query time series, and then use such correlated words as a weighted text query to retrieve documents that match such a query well. This idea naturally leads to a general three-step retrieval procedure as shown in Figure 2. In the first step, we pre-process the text stream and obtain a frequency distribution over time for each term. Such a frequency distribution gives us a time series characteristics when the term is mentioned frequently, and when infrequently, over time. In the second step, we can then match such a term time series with the given query time series to compute a correlation or association score. If whenever the value of a query time series variable is high, a word would occur frequently, we would have a high correlation for such a word. For example, if mentioning of a particular political issue tends to be associated with the increase of votes for a presidential candidate in a national poll, words about such a political issue can be expected to have a relatively high correlation compared with other words. Finally, we would use words with relatively high correlations to compute the score of a document by essentially treating such words as forming a weighted text query.

Clearly, the two main technical challenges in such an approach are: 1) how to measure the correlation between a term and the input time series, and 2) how to aggregate the signals from each term for document ranking. In general, there are many different ways of solving each of these two challenges within the proposed general retrieval framework. In this paper, as a first step in studying this new problem, we focus on studying a few natural ways to solve these challenges, leaving the exploration of more sophisticated methods as future work. Specifically, we will experiment with two methods for computing correlation of two time series variables (i.e., Pearson correlation and Dynamic Time Warping), and two different strategies for computing document scores based on term correlations (i.e., feeding an existing retrieval function with an artificial weighted text query constructed based on correlated terms, and direct aggregation of term correlations). We now describe all these methods in detail.

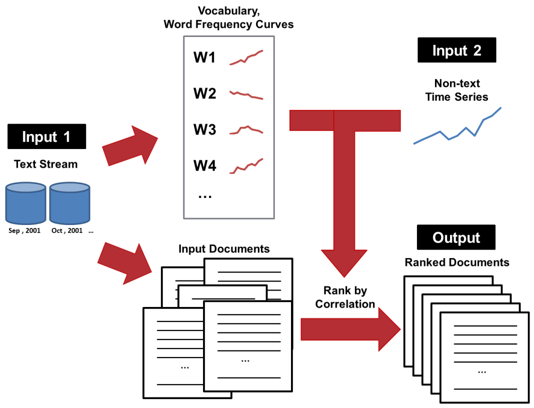


Figure 2: Overview of information retrieval with time series query

4.2 Correlation Function

To compute the correlation of a term with a query time series, we first generate the term frequency curve for each term w in the collection, WF_w, over the given time period of the collection, t_1, ..., t_n.

$$WF_w = (wf_{w,t_1}, wf_{w,t_2}, ..., wf_{w,t_n})$$

$$wf_{w,t_i} = c(w, D_{t_i})$$

where wf_{w,t_i} is the frequency of the word w at the time t_i, and $c(w, D_{t_i})$ is the count of w over all documents d having t_i time tag in collection D.

We can now use any similarity metric for the time series to measure the correlation between a term and the query time series. Depending on the specific applications, we may also be interested in shifting the time points when computing correlations. For example, aligning term frequencies in earlier time points (e.g. 1 day before) to a stock time series can potentially discover terms that might "predict" stock changes, while aligning them from a later time period (e.g. a few days later) than the stock prices might reveal words that discuss the changes of stock that have already happened. In this paper, we do not systematically explore all these options, but instead simply align the time points exactly in order to focus on understanding the relative effectiveness of different methods, though one of the two methods to be presented below can capture the time shift to some extent. Furthermore, we may be interested in assessing correlation in a certain time period, rather than the entire time period. This can also be easily achieved by restricting the computation of correlation to only the interesting time period.

4.2.1 Pearson Correlation

Pearson correlation (**Pearson**) is the most representative metric for measuring how much two time series are related. It indicates whether two time series move in the same direction (an increase or decrease), which fits our purpose very well.

For two time series X=(x_1, x_2, ..., x_N) and Y=(y_1, y_2, ..., y_M), correlation coefficient can be defined as following [2].

$$\rho_{X,Y} = corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

[2] http://en.wikipedia.org/wiki/Correlation_and_dependence

where μ_X and μ_Y are average of X and Y, σ_X and σ_Y are standard deviation of X and Y, E is expected value (average) operator, cov is covariance operator, and corr is correlation.

4.2.2 Dynamic Time Warping

Although Pearson correlation gives a good idea of how much the two time series are correlated, it has a limitation in capturing the similarity when one of the series is stretched or shifted. It is often the case that two time series have overall similar shapes, but are not exactly lined up on the timeline. In reality, when one factor may affect another, there can be a delay in the impact, or this impact may last longer even once the causal factor has disappeared. Therefore, it is possible for the two correlated time series to not be in sync.

To overcome this limitation, dynamic time warping (**DTW**) [17, 22] has been proposed. Dynamic time warping is a dynamic programming algorithm that aligns time series with a flexible timeline mapping. That is, depending on the shape of the series, one time period (e.g. a day) can be mapped to several time periods of the same kind (e.g. days) of the other series dynamically. By following the mapping path from the beginning to the end of the time series, DTW finds the best alignment path with the minimal distance between the time series (e.g. Euclidean distance).

Algorithm starts by building the distance matrix having distances for all pair between X and Y. Each component of cost matrix C is $c_{i,j}$ = $\|x_i - y_j\|$. Then, using the cost matrix, optimal alignment path, p^*, with minimal cost is searched.

$$\begin{aligned} DTW(X,Y) &= c_{p^*}(X,Y) \\ &= min\{c_p(X,Y), p \in P^{N*M}\} \\ &= D(N,M) \end{aligned}$$

where D is accumulated cost matrix. That is, D(i,j) is the minimal cost to align x_1, ..., x_i to y_1, .., y_j. D can be computed with the following rule with dynamic programming.

1. First row: $D(1,j) = \sum_{k=1}^{j} c(x_1,y_k), j \in [1,M]$.
2. First column: $D(i,1) = \sum_{k=1}^{i} c(x_k,y_1), i \in [1,N]$.
3. Other elements: $D(i,j) = \min\{D(i-1,j-1), D(i-1,j), D(i,j-1) + c(x_i,y_j)\}, i \in [1,N], j \in [1,M]$.

As we have mentioned in Section 2, dynamic time warping has also been used in previous works on time series retrieval [32] and word similarity by time dynamics [25].

4.3 Aggregation function

Once we obtain the word correlation scores, our next task is to score documents based on their coverage of highly correlated words. We propose two strategies for doing this. The first one is to formulate a weighted text query using the highly correlated words, and adapt an existing retrieval function to score and rank documents. The second one is to directly compute an aggregated correlation score for a document based on the correlation scores of the words matched in the document. Compared with the first strategy, the second strategy has the advantage of generating a more meaningful score (i.e., correlation score) for our task, but the first one may have the advantage of leveraging existing retrieval heuristics. We now present these two strategies in detail.

4.3.1 Weighted TF-IDF (BM25)

The first strategy is to leverage an existing retrieval model. In our original problem formulation, the query is non-textual, thus we cannot apply a regular retrieval model. However, after we compute word correlations, we can generate a weighted text query based on the highly correlated words. This would allow us to leverage an existing retrieval model for scoring documents. In our experiments, we will use BM25 [10], which is one of the most effective basic retrieval functions. Specifically, we can slightly modify BM25 formula to use a correlation coefficient for each term as a weight. With sorted terms w'_i by correlation coefficient, we can define "top-K term BM25" as

$$score_{TopK-BM25}(d) = \frac{\sum_{i=1}^{min(l,K)} |cor_{TS}(w'_i)| BM25(w'_i,d)}{\sum_{i=1}^{min(l,K)} |cor_{TS}(w'_i)|}$$

$$BM25(w,d) = IDF(w)\frac{f(w,d)(k_1+1)}{f(w,d)+k_1(1-b+b\frac{|d|}{avgdl})}$$

$$IDF(w) = log\frac{N-n(w)+0.5}{n(w)+0.5}$$

where $f(w,d)$ is term frequency of w in d, $|d|$ is document length, $avgdl$ is average document length, N is the total number of documents in the collection, and $n(w)$ is the number of documents having w in the collection. In our experiment, k_1 and b are set as 1.5 and 0.75 respectively which are recommended values in literature [29].

4.3.2 Average Correlation

One potential disadvantage of the Weighted TF-IDF approach is that the scores are not so meaningful. We thus propose a second strategy where we aggregate the correlation values of words matched in a document and obtain an aggregated correlation value, which can be interpreted as the correlation of the topic covered in the document with the query time series.

Average over all terms: A natural way to aggregate word correlations at the document level is to compute the average correlation of all the terms in the document (**AC**). Formally, each document consists of words that form a subset of the entire vocabulary of the input data set, $d = w_1, w_2, .., w_l$. The ranking score of document d can then be defined as

$$score_{AC}(d) = \frac{1}{|d|}\sum_{i=1}^{l}|cor_{TS}(w_i)|$$

where $|d|$ is the length of document and $cor_{TS}(w_i)$ is the correlation between input time series and the word frequency curve of w_i. Note that this aggregation function is independent of the correlation function for terms, thus we can apply it on top of any correlation function. We use the absolute value of the correlation as the weight because we want to focus on the strength of the correlation, regardless its sign. Moreover, we do not want to have two highly correlated terms with different direction that negate each other when being averaged. However, we could also consider the direction of the time series to obtain more detailed scores that can tell which documents had a positive impact or a negative impact on the query time series.

Since stopwords do not provide any meaningful signal about the time series changes, we can preprocess the documents to remove them.

Average over top-k terms: Obtaining a high score for the average correlation requires for all the words in the document to be highly correlated. However, in reality, even if every single word is not highly correlated, there can still exist some important information that could explain the input time series. Besides, a relevant document containing correlated topics may cover non-correlated topics. Based on this intuition, we propose another aggregation method that uses only the top-K highest correlated terms to compute average correlation (**TopK-AC**). That is, documents will compete with each other using their best K terms. Ranking score of the document d can be defined as follows.

For each document, the terms will be sorted in descending order of the absolute values of their correlations. Then, the top K correlations will be used for computing each document's ranking score.

The ranking score of the document d can be defined as

$$score_{TopK-AC(d)} = \frac{1}{|K|} \sum_{i=1}^{min(l,K)} |cor_{TS}(w'_i)|$$

where w'_i are the sorted terms.

Unlike the average correlation, top K average correlation will penalize documents that are shorter than K words, because their score will still be divided by K. This is reasonable, since such short documents are less likely to contain enough information.

Average over top-k unique terms: When we observe only the top K correlated words, the list of their scores may be dominated by multiple occurrences of the same term. In large documents, multiple occurrences are more likely to happen. This may lead to documents that are more diverse to be ranked lower than the less diverse documents where a few highly correlated words are repeated a large number of times. To avoid this, we propose a variation of the top K average correlation where the list of top K correlated words contains only the unique terms (**TopK-AC-Uniq**).

To achieve this, input documents are reprocessed to have each word only once, $d'' = w''_1, w''_2, .., w''_m$. Then, the previous strategy of top K average correlation can be applied.

$$score_{TopK-AC-Uniq}(d) = \frac{1}{|K|} \sum_{i=1}^{min(m,K)} |cor_{TS}(w''_i)|$$

5. EVALUATION

To evaluate our retrieval task, we need a test collection that consists of (1) a text stream, (2) query time series, and (3) relevance judgments for each document with respect to each time series query. Since there does not exist any such test collection that we could use, we had to create one. Below we first discuss how we create our test collection, design our experiments, and then discuss the experiment results.

5.1 Experiment design

Data Set: Considering the availability of many stock price time series data and many news articles of the same time period, we decided to use stock price time series data as queries and the companion news stream as document collection. Specifically, for the text stream, we used The New York Times annotated corpus [3] covering all the news published for the period between July 2000 and December 2001, which includes the total of 144261 articles. For the input time series queries, we have used stock prices of multiple major companies. Daily stock price information was obtained from Yahoo! Finance [4].

Creating relevance judgments turned out to be a challenging task. To find if one document has contents associated with changes in time series, annotators would have to have profound knowledge for both time series and text data. We propose to solve this problem by assuming that a document mentioning the name of a company is relevant to the stock time series of the corresponding company. That is, given a query representing the stock prices of company entity A, we assume that any documents mentioning A are relevant. This gives us an objective way of evaluating our algorithms quantitatively. However, while it is reasonable to assume that if a document mentions entity A, it can be regarded as relevant, not all relevant documents necessarily mention entity A. We thus also examine the top-ranked results and evaluate them qualitatively.

To ensure each query has a reasonably large number of relevant documents in the collection, we selected companies that had more than 50 relevant articles in our document collection and whose daily values of stocks were available on Yahoo! Finance. We ended up obtaining 24 such companies, giving us 24 time series queries with relevance judgments.

Measures: To measure the performance of our algorithms, we used standard information retrieval measures, mean average precision (MAP) and normalized discounted cumulative gain (NDCG) [9]. In our problem setup, we only used the binary relevance, that is, the gain will be either 0 or 1.

Research Questions: First, we want to know if our proposed methods are capable of finding meaningful relevant documents based solely on a non-textual time series query. Second, we want to compare the different variations of our proposed algorithms and understand which method works the best. In particular, we would like to compare the Pearson correlation and the dynamic time warping correlation, and compare the weighted TF-IDF document scoring method with the correlation aggregation methods.

Implementation Detail: For the implementation of dynamic time warping, we have used the Machine Learning Python package (mlpy) [5]. We removed stop words using the stopword list provided by the Python Natural Language Toolkit (NLTK [6]).

5.2 Experiment Results

5.2.1 Qualitative evaluation

We first examine the quality of the top-ranked documents to see if it is feasible to retrieve meaningful relevant documents solely based on a non-textual time series query. In Table 2, we show the dates and excerpts of the top ranked documents obtained by using American Airlines stock price query.

Interestingly, all of the top ranked documents are distributed over the late 2001, and related to the September 11 terrorist attacks and the anti-terrorism activities in 2001. In Table 1, we further show the top-10 most correlated terms to the input time series. We again see that all the top terms are related to terror attack and also confirm that the highly correlated terms indeed contributed to the retrieval of the top-ranked relevant documents in Table 2.

Table 1: Top 10 highly correlated words to AA stock (Pearson)

WORD	$\lvert\rho\rvert$	WORD	$\lvert\rho\rvert$
challenged	0.887031	pakistan	0.848829
afghanistan	0.861351	afghans	0.844596
security	0.858745	afghan	0.843481
sept	0.858309	islamic	0.842499
terrorism	0.854865	taliban	0.841455

Since our relevance judgments are created automatically based on entity recognition, our algorithm can also be used to (automatically) rank stocks based on how strongly their prices are correlated with topics covered in our news stream. Results in Table 4 are ranked by the MAP of Pearson correlation. Results with the American Airline stock query that had a significant cause, namely the September 11 attacks, which was mentioned a lot in the news media, shows the highest precision compared to other companies. This matches our intuition. In general, our algorithm can be used to analyze comparable time series data based on their correlations with topics in text data, offering an interesting novel way of mining text data for interesting topic patterns that might explain changes in time series data and obtaining potentially useful knowledge.

The results of American Airlines stock price query show that our algorithms can effectively cross the barrier of from time-series to text data to retrieval meaningful relevant articles based solely on a time series. We now show that the results can be significantly improved by further imposing a simple keyword constraint. In Table 3, we show the top-10 results of using the stock time series of Apple after imposing the constraint that the documents must contain the term "Apple". Although the Apple stock price has also dropped due to the September 11 attacks, there was an even more significant

[3] `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2008T19`

[4] `http://finance.yahoo.com/`

[5] `http://mlpy.sourceforge.net/`

[6] `http://nltk.org/`

drop at the end of the September 2000, when it turned out that the Apple's earnings report was not as good as its expectation. At that time, the company's stock price made a historical drop (about 50% off). In Table 3, we can observe that all of the top 3 ranked documents directly explain this event. In addition to the earnings report, we can observe the various events that may have affected the company's stock such as the Federal Reserve and the opening of new retail stores. Notice that our algorithm ignored the September 11 related articles, which had a relatively low correlation to stocks in the computer industry.

These results also show that our algorithms can generate an additional ranking score orthogonal to the other ranking features used in a search engine to prefer documents that cover topics correlated with a time series, and thus can help improve the current search engines to generate more useful ranking of documents for the purpose of text analysis in association with a time series.

5.3 Quantitative Evaluation

Table 4: Comparison of Pearson and DTW

QUERY	REL DOCS	Pearson		DTW	
		MAP	NDCG	MAP	NDCG
American Airlines	393	0.0159	0.5143	0.0159	0.5143
Microsoft Corp	575	0.0027	0.449	0.0027	0.449
Ford Motor	440	0.0027	0.4319	0.0038	0.4515
Verizon	128	0.0026	0.3651	0.0013	0.3401
Time Warner	359	0.0026	0.419	0.0041	0.445
Boeing	137	0.002	0.3561	0.0014	0.3453
General Electric	245	0.0017	0.3809	0.0029	0.407
AT&T	273	0.0017	0.3874	0.0026	0.4072
Intel Corp	168	0.0016	0.3601	0.0016	0.3635
Citigroup	157	0.0014	0.3521	0.0009	0.3379
Cisco Systems	154	0.0012	0.3445	0.0016	0.3576
Amazon.com	159	0.0012	0.3455	0.0014	0.3559
Yahoo	126	0.0012	0.3322	0.0011	0.3313
Sony Corp	206	0.0011	0.3554	0.0015	0.3706
Disney	190	0.0011	0.3541	0.0019	0.3771
Hewlett	145	0.001	0.3346	0.0014	0.3495
Goldman Sachs	121	0.001	0.3248	0.0015	0.3429
Coca-Cola	146	0.0009	0.3341	0.0012	0.3437
Apple Computer	89	0.0006	0.2938	0.0006	0.2938
Toyota	107	0.0006	0.3043	0.0009	0.3157
eBay	99	0.0006	0.3008	0.0009	0.3117
Procter&Gamble	84	0.0005	0.2826	0.0009	0.3033
Wal-Mart	77	0.0004	0.271	0.0008	0.2938
Nissan	55	0.0003	0.2428	0.0004	0.2547
Average	193.04	0.0019	0.3515	0.0022	0.3609

Table 4 shows a comparison of the two correlation methods, i.e., Pearson and DTW, by fixing the aggregation function to average correlation (**AC**). We see that the number of relevant documents (from 52 to 575) is very small, when compared to the size of the whole collection (total of 144261 documents). Our algorithm with Pearson correlation shows a 0.0019 mean average precision that is higher than the random precision of 0.0013. It also shows a relatively high NDCG, 0.3515. When we used the dynamic time warping for correlation measure, the results showed an even better performance, with a 0.0022 mean average precision and a 0.3609 NDCG. DTW shows a significant improvement over the Pearson correlation (paired t-test, 95% confidence interval, t-value=2.25 for MAP and 3.43 for NDCG). The evidence shows that the flexible timeline alignment of dynamic time warping helps to better model the real world stock prices.

Table 5 shows the retrieval performance with various aggregation methods when the correlation function is fixed to Pearson correlation. For major comparison, we also performed significant tests with paired t-test with 95% confidence level.

Compared to the average correlation, all variations of the top K average correlation variations and BM25 have shown a significant improvement in NDCG. The results have shown that using only the most correlated terms of the document, instead of all of its terms, results in a more useful ranking. Such technique allows us to avoid the noisy insignificant terms that most documents contain.

Table 5: Comparison of correlation aggregation methods

	MAP	NDCG
AC	0.0019	0.3515
Top5-AC	0.0021	0.361
Top10-AC	0.0023	0.3618
Top20-AC	0.0024	0.3629
Top5-AC-Uniq	0.0022	0.3613
Top10-AC-Uniq	0.0022	0.3616
Top20-AC-Uniq	0.0022	0.3619
Top5-BM25	0.0019	0.3584
Top10-BM25	0.0023	0.361
Top20-BM25	0.0019	0.3582

When we compared the topK to topK-uniq methods, there were no significant differences. For average correlation, when we use a higher K, it shows a better performance for topK-AC methods (significant improvement from top5-AC to top20-AC). We tried larger values of K, the performance may be further improved slightly, but insignificantly. Extremely large values of K hurt the performance. Overall, K=20 seems to be a reasonable value.

BM25-based method showed higher sensitivity to K. It showed the best performance when $K = 10$, which is comparable to the top-K average correlation methods, but its performance for $K = 5$ and $K = 20$ was significantly worse than the top-K average correlation methods. One possible explanation is that the correlation aggregation methods can more directly and likely more accurately quantify the correlation of the topic in a document than using a standard retrieval function such as BM25.

We see that all the MAP values are very low. The main reason for this is because our retrieval task is much more challenging than a normal retrieval task as we face a huge gap between the query (a time series) and a relevant document. Thus the task is much harder than a normal retrieval task on average. Another reason is that our current gold standard does not contain complete relevance judgments, and many top-ranked ones are actually relevant as we have seen in the qualitative evaluation. While we believe the results of relative comparison between different ranking methods are meaningful, the MAP values we obtained are likely a significant under-estimate of the actual MAP values. Using more complete judgments for evaluation is an important future research to be done.

Despite the low MAP values, however, the NDCG values are relatively high. This difference is due to the different discounting coefficients used in the two measures to penalize results with lowly ranked relevant documents. Specifically, MAP penalizes low positions linearly, but NDCG uses a less-harsh logarithmic discounting scheme for penalization. We analyze this difference as follows. Suppose we want to compute $MAP@K$ and $NDCG@K$ for a query with a total of R relevant documents in the collection. Assume that there are n relevant documents returned in the top K results at ranks, $r_1, .., r_n$, respectively. Then, we have

$$MAP@K = \frac{1/r_1 + 2/r_2 + ...n/r_n}{R}$$

$$NDCG@K = \frac{\frac{1}{log(r_1+1)} + \frac{1}{log(r_2+1)} + ... + \frac{1}{log(r_n+1)}}{1 + \frac{1}{log3} + ... + \frac{1}{log(n+1)}}$$

where log is log with base 2. Thus, if we view each as a sum over all the ranks of a relevant document retrieved, then the weight at r_i (rank of the i-th relevant doc) would be i/r_i for MAP and $1/[log(r_i+1)*(1+1/log3+...1/log(n+1))]$ for NDCG. Therefore, the impact of adding 1 to r_i (i.e., moving a relevant down one position in the ranked list), measured by the ratio of the degraded weight to the original weight, would be: $r_i/(r_i+1)$ for MAP and $log(ri+1)/log(ri+2)$ for $NDCG$. We clearly see that $NDCG$ degrades much more slowly.

The difference between MAP and NDCG is further amplified in our results because we computed them over the whole set of docu-

Table 2: Top ranked documents by American Airlines stock price query

RANK	DATE	EXCERPT
1	10/22/2001	FLEEING THE WAR
2	12/11/2001	US and anti-Taliban forces in Afghanistan
3	11/18/2001	Fate of Taliban Soldiers Under Discussion
4	11/12/2001	Tally of dead and missing in Sept 11 terrorist attacks (S)
5	11/12/2001	Western soldiers in Afghanistan with Northern Alliance troops near city of Taliqan
6	9/25/2001	Recovery operation at World Trade Center following Sept 11 terrorist attacks
7	11/19/2001	Officials estimate that as of Friday, 4,343 people had died, or were missing and presumed dead, as a result of the attacks on Sept. 11. AT THE WORLD TRADE CENTER 3,953 dead or missing 636 confirmed dead, with 594 identified 157 dead on two hijacked planes, including 10 hijackers.
8	11/3/2001	Dead and Missing report of Sep 11 attack
9	11/17/2001	Dead and Missing report of Sep 11 attack
10	11/18/2001	Dead and Missing report of Sep 11 attack

Table 3: Top ranked relevant documents by Apple stock price query

RANK	DATE	EXCERPT
1	9/29/2000	Apple Computer says its fourth-quarter earnings will fall far below analysts' estimates ...
2	12/8/2000	Apple Computer; company has about $4 billion in cash and short-term investments in reserve, not $11 billion
3	10/19/2000	Apple Computer announces earnings in fiscal fourth quarter ended Sept 30 that narrowly miss Wall Street's lowered estimates
4	4/19/2001	Dow and Nasdaq Soar After Rate Cut by Federal Reserve
5	7/20/2001	Apple Computer's new retail stores regarding history of company's relationship with several retailers
6	12/6/2000	Apple Warns It Will Record Quarterly Loss
7	3/24/2001	Stocks Perk Up, With Nasdaq Posting Gain in a Harsh Week
8	8/10/2000	Mixing Mac and Windows
9	1/18/2001	Apple Posts $247 Million Loss for Quarter
10	6/10/2001	... corporate alliances unravel and dissolve within decade of inception; prompts consideration of other business alliances, including those between .... Apple and Motorola;

ments. Normally, in retrieval evaluation, we compute NDCG@10 because users generally only look at the top 10 documents, and any differences afterward probably do not matter to our users. However, in our evaluation, our goal is to compare two ranking methods for a very difficult ranking task, we did not use any cutoff and simply used the entire set of documents in order to see any small differences between different methods.

6. DISCUSSIONS

Although we focused on studying the accuracy of the proposed algorithm for solving this new problem in this paper, in practical applications, we would also need to consider the efficiency of the algorithm. Our envisioned application of the proposed retrieval problem is not for interactive search with time-series queries, but rather for joint analysis of text data and time-series data to discover topics that can potentially explain the changes of the time series data. Thus it does not have to respond in real time as in a regular Web search engine. For example, when analyzing stocks, we can easily afford to spend a few hours to find the most relevant topics. However, it is also possible to imagine other applications where faster responses are highly desirable. In such a case, there are several strategies to consider. First, when we save text collection, as a preprocessing step, we can generate word frequency time series for each term in the data set and index them by terms. Second, our proposed methods can be easily parallelized. When an input time series query arrives, we can calculate correlation coefficient with each term in parallel because the computation for each term is independent. The aggregation step can also be easily parallelized by aggregating a subset of data. The correlation coefficient computation may even be pipelined to the aggregation step for more speedup.

However, because the vocabulary size is huge, the correlation computation may still take a long time. To solve this problem, we can reduce dimensions by clustering temporally and semantically similar terms into a cluster. By using precomputed cluster time series as a unit, we would be able to decrease the amount of computation significantly in the correlation coefficient step as well as the aggregation step.

We assumed that 'relevant documents to time series' are highly correlated documents to the input time series that can potentially cause or be caused by the change of time series. Depending on what kind of correlation function to use, users may capture different notions of relevancy thus flexibly supporting different kinds of applications. For example, if a user is only interested in unidirectional causality from a document to input time series, it is possible to use a causality test with lagged value such as Granger test [7] as a correlation function.

In our evaluation, we assumed a document mentioning a company name to be relevant to the query representing the corresponding company's stocks. While this assumption makes it easy to perform quantitative evaluation, it also has its limitation. In particular, it appears that many relevant documents with topics related to the query time series do not necessarily mention the company's name. In our qualitative evaluation, we found that many top-ranked documents are clearly relevant, but they did not mention the company's name. As a result, they were regarded as non-relevant. This may also be a factor that has artificially lowered the MAP values. Further evaluation with more complete relevance judgments is thus needed in order to draw more reliable conclusions. However, we would like to stress that our results clearly demonstrate the usefulness of the proposed new retrieval task in helping an analyst to go from an arbitrary time series query into a text collection to explore relevant and related topics in the text data, providing a novel way to support text analysis.

7. CONCLUSIONS

We introduced and studied a novel retrieval problem with time series queries, where the goal is to find relevant documents in a text collection of the same time period as the time series query, which contain topics that are correlated with the query time series. This retrieval task is very useful for a user to analyze a time series jointly with companion text data, particularly in helping users understand the changes of a time series.

To solve this problem, we proposed and studied multiple retrieval algorithms that use the general idea of ranking text documents based on how well their terms are correlated with the query time series. Experiment results show that it is feasible to use a non-textual time series query to directly retrieve relevant text documents, and the proposed retrieval algorithms can effectively find relevant documents that are useful for understanding the changes of the query time series.

Our study only explored some basic algorithms for solving this problem, but the proposed solution framework is quite general and can thus easily support further exploration of this problem with more sophisticated correlation functions and aggregation functions, opening up many interesting directions for future research in this direction. First, we may extend the current algorithms with other advanced information retrieval techniques, such as query expansion and feedback process. For example, we may retrieve the top N highly correlated terms and use them as a supplementary measure of the input time series to find correlated documents. Second, we may perform topic analysis to capture the relationships among the terms and retrieve correlated topics with a time series. In this paper, we assumed independence of terms and computed the correlation between each term and the input time series independently. A potentially more interesting way may be to group related terms together (e.g. via semantic similarity, often co-occurring terms, or a high temporal correlation between terms) and measure correlation between the groups of terms (i.e., topics) and the input time series. Lastly, our quantitative evaluation relied on highly incomplete relevance judgments, thus a very important future work is to generate a test set by using experts to create a more complete set of relevance judgments, and further verify the effectiveness of the proposed methods with more solid evaluation.

8. ACKNOWLEDGMENTS

This material is based upon work supported in part by the National Science Foundation under Grant Number CNS-1027965 and by an HP Innovation Research Award.

9. REFERENCES

[1] R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient similarity search in sequence databases. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, FODO '93, pages 69–84, London, UK, UK, 1993. Springer-Verlag.

[2] R. Agrawal, K.-I. Lin, H. S. Sawhney, and K. Shim. Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB '95, pages 490–501, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[3] R. Agrawal, G. Psaila, E. L. Wimmers, and M. Zaït. Querying shapes of histories. In *Proceedings of the 21th International Conference on Very Large Data Bases*, VLDB '95, pages 502–514, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[4] K.-P. Chan and A. W.-C. Fu. Efficient time series matching by wavelets. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 126 –133, mar 1999.

[5] M. Efron. Linear time series models for term weighting in information retrieval. *J. Am. Soc. Inf. Sci. Technol.*, 61(7):1299–1312, July 2010.

[6] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM international conference on Management of data*, SIGMOD '94, pages 419–429, New York, NY, USA, 1994. ACM.

[7] C. W. J. Granger. Essays in econometrics. chapter Investigating causal relations by econometric models and cross-spectral methods, pages 31–47. Harvard University Press, Cambridge, MA, USA, 2001.

[8] D. Gunopulos and G. Das. Time series similarity measures. In *Tutorial notes of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 243–307, New York, NY, USA, 2000. ACM.

[9] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, Oct. 2002.

[10] K. S. Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Inf. Process. Manage.*, 36(6):779–808, Nov. 2000.

[11] R. Jones and F. Diaz. Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25(3), July 2007.

[12] J. Kacprzyk and A. Wilbik. Linguistic summarization of time series using linguistic quantifiers: Augmenting the analysis by a degree of fuzziness. In *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on*, pages 1146 –1153, june 2008.

[13] M. Kaya and M. Karsligil. Stock price prediction using financial news articles. In *Information and Financial Engineering (ICIFE), 2010 2nd IEEE International Conference on*, pages 478 –482, sept. 2010.

[14] E. Keogh. Fast similarity search in the presence of longitudinal scaling in time series databases. In *Proceedings of the 9th International Conference on Tools with Artificial Intelligence*, ICTAI '97, pages 578–, Washington, DC, USA, 1997. IEEE Computer Society.

[15] E. Keogh and P. Smyth. A probabilistic approach to fast pattern matching in time series databases. page 24. AAAI Press, 1997.

[16] E. J. Keogh and M. J. Pazzani. Relevance feedback retrieval of time series data. In *Proceedings of the 22nd annual international ACM conference on Research and development in information retrieval*, SIGIR '99, pages 183–190, New York, NY, USA, 1999. ACM.

[17] E. J. Keogh and M. J. Pazzani. Derivative dynamic time warping. In *In First SIAM International Conference on Data Mining (SDM'01*, 2001.

[18] H. D. Kim, C. Zhai, T. A. Rietz, D. Diermeier, M. Hsu, M. Castellanos, and C. A. Ceja Limon. Incatomi: integrative causal topic miner between textual and non-textual time series data. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 2689–2691, New York, NY, USA, 2012. ACM.

[19] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais. Understanding temporal query dynamics. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 167–176, New York, NY, USA, 2011. ACM.

[20] R. Liebscher and R. K. Belew. Lexical dynamics and conceptual change: Analyses and implications for information retrieval. *Cognitive Science Online*, pages 46–57, 2003.

[21] C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[22] M. Meinard. *Introduction to Information Retrieval*. Springer, 2007.

[23] G. Mitra and L. Mitra. *The handbook of news analytics in finance /*. Wiley ;, Hoboken, N.J. :, 2011.

[24] M. Ng and Z. Huang. Temporal data mining with a case study as astronomical data analysis. *Lecture Notes in Computer Sciences*, pages 2–18, 1997.

[25] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 337–346, New York, NY, USA, 2011. ACM.

[26] D. Rafiei and A. Mendelzon. Similarity-based queries for time series data. *SIGMOD Rec.*, 26(2):13–25, June 1997.

[27] D. Rafiei and A. Mendelzon. Efficient retrieval of similar time series. In K. Tanaka, S. Ghandeharizadeh, and Y. Kambayashi, editors, *Information Organization and Databases*, volume 579 of *The Springer International Series in Engineering and Computer Science*, pages 75–89. Springer US, 2000.

[28] R. P. Schumaker and H. Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Trans. Inf. Syst.*, 27(2):12:1–12:19, Mar. 2009.

[29] A. Singhal. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–42, 2001.

[30] K. Takahashi and M. Umano. Retrieval of similar time series with similarity degree of linguistic expressions for global trend and local features. In *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*, pages 1 –8, june 2012.

[31] M. Umano, M. Okamura, and K. Seta. Improved method for linguistic expression of time series with global trend and local features. In *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*, pages 1169 –1174, aug. 2009.

[32] B.-K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proceedings of the Fourteenth International Conference on Data Engineering*, ICDE '98, pages 201–208, Washington, DC, USA, 1998. IEEE.

[33] C. Zhai. *Statistical Language Models for Information Retrieval*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2008.