Extracting Temporal Patterns and Analyzing Peak Events

K.R. Premlatha, T.V. Geetha

Department of Computer Science & Engineering, College of Engineering, Guindy, Anna University, Chennai, India.

Abstract—Temporal text mining (TTM) is the process of discovering sequential and temporal patterns in text information collected over time. This is useful in application domains where each entity of text in a text stream (usually a document or publication) has a meaningful timestamp. In this paper, extraction, formalization and comparison of temporal terms has been performed. The extraction of temporal expressions (explicit, implicit, and vague) has been performed by using FSA. In formalization, the natural language expression is converted to calendar based time-line. Vague expressions have been handled based on the reference time and the duration of indexical features. Peak event has been analyzed by using the start-end time of the event and reference article for the particular event. Start-end time of the particular event has been used as timestamp while analyzing peak event. Finally temporal similarity between events has been calculated and used to convey temporal relation between documents. This similarity has been calculated using Allan’s approach. The similarity has been shown as document event matrix conveying temporal relations. This matrix also highlights the peak events across the documents. Hot topic of the documents has also been exposed while comparing temporally similar events.

Index Terms—text mining, temporal similarity, clustering, peak event

I. INTRODUCTION

Temporal Text Mining (TTM) focuses on determining temporal patterns in text information that are composed over time. The stream of text documents hold relative time information based on their arrival over time and their growth through interactive time-dependent processes. News documents contain a wealth of information coded in natural language temporal expressions. Automatic processing of news often ignores these expressions for several reasons: temporal expressions are difficult to spot, their surface forms themselves are hardly of any direct use, the meaning of an expression is often somewhat ambiguous or at least very difficult to resolve, and the terms do not lend themselves easily to any kind of evaluation. However, temporal expression would be highly useful for many areas of application: information extraction, information retrieval, question answering, document summarization and topic detection and tracking [1]. Many information resources have a stream-like formation, in which the way content arrives over time carries an important part of its meaning. The proliferation of on-line information sources and on-line forms of communication has led to various examples: e-mail, chat, discussion boards, and blogs all represent personal information streams with complex topic modulations over time. Analyzing the temporal characteristics of these types of information streams is part of the broader area of sequential pattern mining within the field of text mining, and can be viewed as an application of time-series analysis, a fundamental area in statistics [13].

There are three things one has to deal with before temporal expression can be used in any of these tasks: extraction, formalization, and comparison of the temporal expressions. For extraction temporal expression in the text one often needs some background information, such as the reference time and tense of the relevant verb, in order to map the expression temporally [2]. There has to be an appropriate methodology to represent the temporal expression in a formal manner [7]. Finally, outline an approach to comparing the temporal similarity of two documents. Need to model the temporal similarity in terms of overlie in the temporal references by running pair wise comparisons of documents. The result of comparison characterizes the proportion of overlap in the determined expressions of two documents [2].

An important aspects that can be considered in information resources with stream like structure is peak
event or hot topics analysis. Peak event or hot topics are basically analyzed by arrival time of the documents [6] and formation of clusters [8]. Cluster-based information retrieval organizes the document collection into groups of closely associated documents [3]. Several methods have been used for cluster formation: Hierarchical methods, Partitioning methods, Density-based Methods, Grid-based Methods, Model-based Methods [9].

This paper is organized as follows: section 2 presents extraction of temporal expressions; section 3 presents the calendar based time-line for converting the natural language expression in to calendar model; section 4 presents the peak event analysis; section 5 presents the temporal similarity of events; finally, section 6 presents conclusion.

II. EXTRACTING TEMPORAL PATTERNS

Extracting temporal patterns from text requires handling of explicitly, implicitly or vaguely expressed temporal information. Some temporal expressions are explicit [4], e.g., on the 18th of March 2009, in November 2010 etc. No additional information is necessary in extracting temporal information. Some other expressions are implicit expressions. These all contain a varying degree of indexical attributes: last Sunday, three weeks ago, on Monday etc. In this case we should know the reference time and the verb tense. Finally vague expressions are before July, after several weeks etc.

Table 1. Temporal term categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseterm</td>
<td>day, week, weekday, month,</td>
</tr>
<tr>
<td></td>
<td>montname, quarter, season,</td>
</tr>
<tr>
<td></td>
<td>year, decade</td>
</tr>
<tr>
<td>indexical</td>
<td>yesterday, today, tomorrow</td>
</tr>
<tr>
<td>internal</td>
<td>beginning, end, early, late,</td>
</tr>
<tr>
<td></td>
<td>middle</td>
</tr>
<tr>
<td>determiner</td>
<td>this, last, next, previous,</td>
</tr>
<tr>
<td></td>
<td>the</td>
</tr>
<tr>
<td>temporal</td>
<td>in, on, by, during, after,</td>
</tr>
<tr>
<td></td>
<td>until, since, before, later</td>
</tr>
<tr>
<td>post modifier</td>
<td>of, to</td>
</tr>
<tr>
<td>numeral</td>
<td>one, two . . .</td>
</tr>
<tr>
<td>ordinal</td>
<td>first, second . . .</td>
</tr>
<tr>
<td>adverb</td>
<td>ago</td>
</tr>
<tr>
<td>meta</td>
<td>throughout</td>
</tr>
<tr>
<td>vague</td>
<td>some, few, several</td>
</tr>
<tr>
<td>recurrence</td>
<td>every, per</td>
</tr>
<tr>
<td>source</td>
<td>from</td>
</tr>
</tbody>
</table>

In this paper, Finite State Automata has been used for extracting all three types of temporal expressions (explicit, implicit and vague). Temporal term categories listed by Juha Makkonen [2] has been used as indicative clues (given in Table 1) for temporal expression extraction. Extraction of vague expression has been carried out by determining the coverage of the interval conveyed in a vague manner.

For example after several weeks covers an interval of several quantities of points on the time-line, some more vaguely than other. However, difficulty arises while formalizing the vague expressions which will be discussed in the next section.

Initially an FSA to extract temporal expressions with year and month as base term is explained as shown in Figure. 1. The input contains a natural language sentence that is scanned a word at a time. The automata remain in the primary state unless a temporal to the primary state. The applicable end states enclose double circles. However this automata in addition also extracts vague temporal expressions. Two examples of sentences with vague temporal expressions is given below.

Example for Extracting Vague Expression:

“After several years of rapid growth, the health care company became one of the largest health care providers in the metropolitan area.”

“Moss has continued to collaborate with Topshop. Her most recent collection was released in May 2010. Since the April 2007 launch, the Kate Moss for Topshop line has expanded to include lingerie and sleepwear.”

The Finite State Automata for year also covers vague terms. By extracting expressions indicating temporal as well as vague categories, the FSA will tackle vague expression such as After several years, Since the April 2007 are extracted.

III. CALENDAR BASED TIME LINE
In order to formalize the structure of the timeline, the calendar model given by Goralwalla et al [5] construct algebra for temporal expressions has been adopted. The use of calendar based modal enables recognition of date and duration of the events [7]. Calendar based model has been used to formalize explicit and implicit temporal expressions based on the canonization approach [2, 3]. Canonization of explicit and implicit temporal expressions has been carried out [2, 3]. The work discussed in this paper also handles formalization of vague expressions. This essentially entails the conversion of vague expressions into date format with duration to cater to the calendar model.

**Canonization**

Mapping the temporal expressions onto a calendar is called as canonization [2], since it determines the formal meaning of a temporal expression.

- a time-line - points with precedence relation,
- a set of granularities (year, month, week, . . )

The terms are mapped as periods \([t_{start}, t_{end}]\) of the bottom granularity which in our case is day.

- A function \(\pi(G, t) = t_i\) \(\rightarrow\) returns the previous start point of an element of the granularity \(G\).
- A function \(\rho(G, t) = t_i\) \(\rightarrow\) returns the start point of the next element of the granularity \(G\).
- Example- reference time \(t_i = 20100728\) that is July 28th 2010.

\[
\begin{align*}
\pi(G_{\text{week}}, t) &= 20100726 \\
\pi(G_{\text{august}}, t) &= 20090801 \\
\rho(G_{\text{year}}, t) &= 20110101 \\
\rho(G_{\text{naughty}}, t) &= 20100803
\end{align*}
\]

- “three months ago” \(\gamma(G, n, t) = t_i\)
- \(\gamma(G_{\text{month}}, 3, t) = 20100428\)

**Table 2. Canonized Expressions with respect to July 28, 2010**

<table>
<thead>
<tr>
<th>No.</th>
<th>Prefix</th>
<th>Base Term</th>
<th>Postfix</th>
<th>Start</th>
<th>End</th>
<th>Type of expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>On the 22(^{nd})</td>
<td>(day)</td>
<td>of October last year</td>
<td>20091 022</td>
<td>20091 022</td>
<td>Explicit</td>
</tr>
<tr>
<td>2.</td>
<td>In late</td>
<td>May</td>
<td></td>
<td>20100 524</td>
<td>20100 531</td>
<td>Implicit</td>
</tr>
</tbody>
</table>

In formalizing vague expressions, there is a need to calculate the duration of indexical features for every point of the time-line. The vague terms few, some, several are associated with duration based on the base terms. For example few weeks ago covers the duration of two weeks approximately while some months ago covers the duration of three months approximately. Table 2 shows the Canonized temporal expressions for explicit, implicit and vague temporal expressions with respect to reference time July 28, 2010.

The natural language expression is converted into calendar based time-line that is in Date format. The Start-End pairs of events of all documents is found based on a given reference time. The next step is the analysis of the peak event.

**IV. PEAK EVENT ANALYSIS**

Some of the significant tasks handled by text mining are: trend analysis, document clustering, deviation detection and discovery of rules of association. There is a method for analyzing news with the aim of discovering a special kind of association between the topics reported over a time span [14]. Peak event and hot topics are analyzed based on the arrival time of the document or article [6].

In this paper, peak event has been analyzed by using the timestamp of start-end time of the particular event. Before peak event is analyzed, a topic based clustering of the calendar line tagged documents is carried out. The general preprocessing used for clustering such as stop word removal, stemming is performed and the document is then represented using \(tf.idf\) vector representation [8] which is then used for clustering. Clusters are formed by using cosine similarity of vectors of the text documents [9]. Cosine similarity between all vectors has been calculated by using the following formula.
The document event matrix shows the peak event for the duration Feb 2010 to Apr 2010. The peak event is Chennai Super Kings IPL match that is highlighted. The sample input documents are related to IPL match, BSNL strike, Expired medicines, Sania mirsa marriage, etc.

VI. CONCLUSION

In this paper explicit, implicit and vague expressions are extracted and formalized into the calendar model which in turn enables the comparison of temporal intervals conveyed by these expressions. The duration associated with these expressions are then used for peak event analysis. The main contribution of this paper is in formalizing vague temporal expressions using calendar model, and use of this model in finding peak event and in measuring temporal similarity of events and peak event analysis based on the timestamp of start-end event duration. Another contribution of this paper is the document event matrix built to convey temporal relations between events across documents. As future work, direct and inverse relationship of two topics and burst topics can be identified from the stream of news corpus.

REFERENCES


