A Framework for Temporal Information Search and Exploration

Parul Patel and S.V. Patel

Abstract Volume of digitized Information is growing drastically on web, digital libraries and other archives. Demand for searching a relevant document or data of specific time period over large amount of data has also increased. Therefore, Time dimension has its own importance in any information domain. Despite of the importance of temporal data available in the document, current search engines and searching techniques provide limited search facilities using date of timestamp or document publication date. Existing retrieval models do not take advantage of *temporal expressions* embedded into a document. This paper describes our framework to exploit temporal expressions in documents in order to add value to the existing information retrieval systems by providing searches like "before elections 2014," "after Diwali," etc., and retrieve relevant documents satisfying temporal expression search criteria.

Keywords Temporal information retrieval · Temporal search time-based clustering

1 Introduction

Web is growing with digitized document where search is an important activity to get required information from large amount of data. Search engine is one of the biggest tools to be used by everyone around the world. Search engine is a kind of information retrieval system that asks user for a specific query and return a list of ranked URL, or documents with their titles and summary of web page or document.

P. Patel (✉)
M.Sc (I.T) Programme, VNSGU, Surat, Gujarat, India
e-mail: parul.patelns@gmail.com

S.V. Patel
Department of Computer Science, VNSGU, Surat, Gujarat, India
e-mail: patelsv@gmail.com

S.C. Satapathy et al. (eds.), *Proceedings of International Conference on ICT for Sustainable Development*, Advances in Intelligent Systems and Computing 409, DOI 10.1007/978-981-10-0135-2_38

In some search engine, facility is provided to search between specific time period by allowing user to enter start and end date into input box and then sorting a retrieved results as per user specified chronological order. But queries like "Elections in India before 2000" requires proper treatment of temporal expressions embedded into a user's query. In above example, user is interested into a document stating information about election before year 2000. So as a result, all documents containing information related to election before the year 2000 must be returned. Another Example, someone who is new to India wanted to know about Indian politics and moreover interested in knowing about "Anna Hazare". In this example, user is interested in knowing details of "Anna Hazare" in chronological order like Anna in 1990, Anna in 1991, etc. A simple query like "Anna Hazare" will not satisfy that requirement. User has to give query like "Anna Hazare from 1960 to 2015." Existing Search engine are not able to handle such queries where temporal expression is leveraged. Moreover, existing retrieval model do not take benefit of temporal expressions contained into the documents.

This paper presents a framework to overcome with above limitations by adding new functionalities to use temporal expression embedded into the documents to utilize them into retrieval. It also handles temporal expression into user's query.

The paper is organized as follows: Sect. 2 presents literature survey on temporal information processing and time based retrieval models. In Sect. 3, Research methodology that includes framework of temporal information retrieval with components such as our temporal tagger, process to retrieve document based on time and an algorithm to represent the retrieved documents on timeline is described in Sect. 3. In Sect. 4, Results and Evaluation of system is presented. Section 5 concludes the paper and gives direction for the future work.

2 Related Work

Developing Framework for temporal information retrieval focus on two different area: (1) Temporal Information Extraction and Processing (2) Use of Such expression in Exploration of search results. Our Literature survey focuses on research that has been done in both of this area. First, we have described research that has been done in development of temporal tagger in various languages. Second phrase is a literature survey about work that has been done in temporal information retrieval.

The Message Understanding Conferences (MUCs) in 1996 and 1998 have played a significant role, but their evaluations covered only recognition of TEs, while a novel contribution towards the normalization of TEs was made in 2000 [1]. GUTime was a rule based system which was developed an extension of TempEx tagger. It was based on TimeML TIMEX3 format, which allows a functional style of encoding offsets in time expressions. It was evaluated on TERN 2004 corpus and achieved 85 % of F-measure [2]. Llorens has developed temporal information extraction system based on CRF for Spanish documents with F-measure of 91 % [3]. KUL is a machine

learning-based system for recognition and normalization of temporal expression with 0.85 % precision and recall of 0.84 % [4]. Negri and Merseglia has developed a rule based system which involves tokenization, part-of-speech tagging based on a list of 5000 entries retrieved from WordNet. Then, the recognized text is processed by a set of approximately 1000 basic rules. Recognized temporal expressions and information around that is used for normalization. Then composition rules are used to resolve ambiguities wherever multiple tag placements are possible. The results in terms of F-measure on ACE 2004 data are 92.6, 83.9 and 87.2 % for detection, recognition and determining the VAL attribute value, respectively [5]. Heideltime is high quality rule based tagger for temporal expression recognition and normalization with 0.90 % precision and 0.82 % recall [6]. The Yamcha is machine learning based tagger which uses SVM and FOIL for chunking and classification of chunks. They got precision of 80.05 %, recall of 73.71 % and F-measure of 76.75 %. They have concluded that use of SVM leads to overfitting [7]. Jelena has developed a system for temporal information extraction and interpretation for serebian language with precision of 0.93 %, recall of 0.96 % and F-score or 0.94 % [8]. SUtime is the library for recognizing and normalizing temporal expressions developed by Stanford University. It is rule-based system developed in java [9].

Research has been done in extracting and processing temporal information from document in various languages like English, Hindi, Spanish, Chinese, etc., but less efforts are made in using that processed data for retrieval and presentation of the document. Research paper on the special issue on temporal information processing by Mani gives road map in this domain. It also focuses on challenges and opportunities in this domain [10]. Google has also added a prototype view:timeline() to display search result on timeline [11]. Xiaoyan Li and Croft has proposed Time bases language models which incorporate time into both query likelihood language models and relevance based language models [12]. Temporal mining of blogs is presented in [13]. J. Allen and R. Gupta and Khandelwal has proposed methods to construct temporal summaries of news stories [14]. Ricardo Baeza Yates has developed an algorithm to obtain future possible events and then searching those events for future information needs [15]. SNAKET is a system developed by paolo and Antonio for unifying hierarchical web snippet clustering with a web interface for web search, books, news and blog domains [16]. Rosie Jones and Diaz have focused on constructing query specific temporal profiles based on publication time of relevant document [17].

Various temporal taggers have been developed to extract and normalize temporal expressions from the document. However, these taggers mainly focus on Explicit temporal expressions hence they extract very few implicit temporal expressions. It may be observed that some documents, we may have large number of implicit temporal expressions like “last diwali,” “next holi,” etc. In such cases our objective is to develop a temporal tagger which extract all Indian festivals as well as of other temporal expressions from document and normalize it into a specific value. By developing such tagger, we have used it into development of our framework for temporal information retrieval and presenting retrieved document into time lined manner.

3 Research Methodology

3.1 Time, Temporal Expression and Temporal Tagger

Time is very important dimension in any information retrieval system. Temporal information is present into the form of temporal expression in any document. Processing such temporal expression from raw text is fundamental requirement for application like text summarization, question answering. A temporal expression also known as Timex also refers to every natural language phrase that denotes a temporal entity like interval or an instant. For example, "Prime Minister Narendra Modi will visit China tomorrow," "India won the test match on last Friday."

Temporal expressions can be classified into following categories according to Schilder and Habel [18].

Explicit Date Expressions such as "13/08/2013", "15th August" refer explicitly to entries of a calendar system and can be mapped directly to temporal Chronons in a timeline.

Implicit All temporal expressions that can be evaluated via a given time ontology and capability of the named entity extraction approach such as name of holiday (last chritmas), next valentine day, etc.

Relative Some temporal expressions express vague temporal information and it is rather difficult to precisely place the information expressed on a time line. Such temporal expressions can be only anchored in a timeline in reference to another explicit or implicit already anchored temporal expression. For example, "on Monday," "Before June and After March," etc. If the document has creation date, then they can be easily anchored. Such reference date can be used to map with chronon and can be used during normalization.

We have developed our own rule based temporal tagger to extract temporal expression from document and normalize in into some standard format. First we have extracted all temporal expressions from the document, then all temporal expressions are normalized into standard values based on offset and reference date. Our tagger has one important characteristics compared to other temporal tagger that it supports normalization of Indian festivals which do not occur on some fixed days. It can handle temporal expression like "last diwali" and can translate into specific date based on selected reference time. First we have extracted temporal reference date and then tried to normalize all temporal expression by considering this reference date. We have stored data of 50 years of Indian festivals into dataset because all Indian festivals do not occur on fixed date. Our tagger is generalized to incorporate new festivals, and with new values of coming year for existing festivals. It also allows incorporating some special events like "tsunami," "attack on taj," etc.

3.2 Temporal Outline of Document

Based on the extracted temporal expressions and their respective normalized values, temporal outline of the document is generated. Temporal Outline can be defined as:

$$\text{TOD} : D \rightarrow [t \times n \times d \times m \times y \times p]$$

where t is a set of temporal expressions extracted from documents.
n is a respective normalized value of temporal expression
m is month chronon,
y is year chronon,
d is date chronon, and
p is a position of the temporal expression into the document.

We can have much temporal expression in the document. So D can be a collection of

$$\{(t_1 \times n_1 \times d_1 \times m_1 \times y_1 \times p_1)\,(t_2 \times n_2 \times d_2 \times m_2 \times y_2 \times p_2).\ldots\ldots\ldots(t_r \times n_r \times d_r \times m_r \times y_r \times p_r)\}$$

where r is number of temporal expressions into the document.

Temporal outline of the document makes all temporal expressions from the document explicit for the further processing.

3.3 Exploring Search Result on Timeline

In the following section, we describe our algorithm to explore search result on timeline.

```
Input : User Query
Output: List of Documents arranged in Timeline Manner

Begin
Step 1:     Parse User query to Temporal Tagger
Step 2:     If Query contains Temporal Expressions
                  search based on the keyword + Temporal
                  Expression (e.g Query is :election on
                  last  Christmas then search applied on
                  keyword Diwali+ 25/12/XXXX+christmas)
            else
                  Search based on query (e.g elections)
            end if
            End
```

Let R is collection of retrieved document on specific user query. We assume that each document has unique id. Following algorithm is used to generate timeline.

```
Begin
Step 1 : Select Smallest and Largest Temporal Chronon
         form selected document's temporal
         outline(tod)
    Chmin(R)= Chminimum(chmin(d1), chmin(d2), chmin(d3)
              ……..chmin(dn))
    Chmax(R)= Chmaximum(chmax(d1), chmax(d2), chmax(d3)
              ……..chmax(dn))
Step 2:   Based on chmax and chmin upper bound and
          lower  bound of timeline is decided
Step 3:   If    IssameGranularity(getgranularity(Chmax),
                getgranularity(Chmin)) is same
                      granularity= getgranularity(chmax)
          else
              granularity=getgranularity
                          (coarsegranule(chmax, chmin))
          end if
Step 4:  Initialialize clusters based on granularity
         seleceted.
Step 5:   If issameyear(chmax,chmin)
                Generate level i of Timeline for 12
                months
          else
                if issamemonth(chmax, chmin)
                      Generate Level i of Timeline for
                      weeks
                Else
                      if issameweek(chmax, chmin)
                            Generate Level i of Timeline
                            for days
                      end if
                end if
          end if
Step 6: Repeat step 5 till documents are there with
        finer granule available into collection R

End
```

Once the upper and lower bound of timeline is fixed, it classification of each document based on their temporal values stored into TDO needs to be done. Each Cluster in timeline contains documents belonging to that chronon. Each document may contain more than one temporal expression, so their TDO may contain more than one value. So it is obvious that that document may belongs to more than one cluster. It may be possible that some clusters do not find any document belonging to that chronon. We have finally revised timeline by removing such clusters from timeline. Once all clusters are initialized with their corresponding links, it is sent to user interface. Each Cluster can be refined into smaller chronon by user if documents have finer granules available into temporal document outline.

4 Evaluation

The initial step was to annotate document by time. From The Times of India archive of different time period, we extracted 100 news documents based on key word "Elections." All these documents were processed using our temporal tagger. The extracted temporal expressions were stored into database with their normalized values and position into a document. Through web interface we queried like

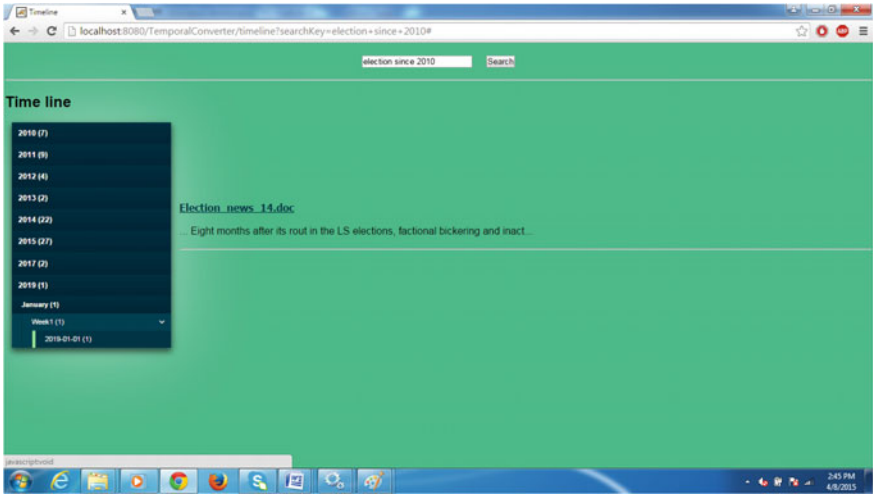

Fig. 1 TimeLine for user query "election since 2010"

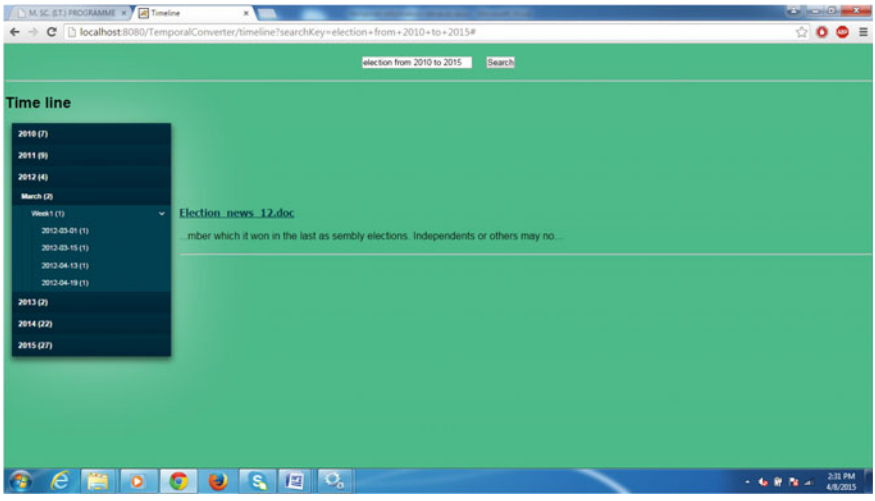

Fig. 2 Search results based on query "election from 2010 to 2015"

"Election from 1990 to 2000," "Election before this diwali," "election after this diwali," "election," "election since 2000," etc. Each document in the respective cluster was checked manually and compared with the respective values. There were 90 % relevant documents into each cluster. Following snapshots show the output of above queries (Figs 1 and 2).

The result is quite satisfactory to use the system for temporal information retrieval supporting temporal expressions (Fig 2).

5 Conclusion and Future Work

Temporal expressions are important structures available into a document and can be useful to improve traditional search technique. We discussed our temporal tagger which not only recognize temporal information available into document, but also normalize it into some standard form such that it becomes explicit for use in other applications. The framework developed can be used to utilize temporal information embedded into document for retrieval of documents and to make time based search and to explore search results on the timeline manner to make visualization more effective. In future, ranking algorithms can be applied on documents in each cluster when many documents are there of same granule. We are working further to improve accuracy as well as doing ranking of documents in individual cluster when many documents are there of same granule.

References

1. Wilson, G., Mani, I., Sundheim, B., & Ferro, L. (2001). A multilingual approach to annotating and extracting temporal information. In *Proceeding of workshop on temporal and spatial Information Processing* (Vol.13, pp. 1–7).
2. Mani, I., & Wilson, G. (2000). Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (Hong Kong)* (pp. 69–76).
3. Llorens, H., Saquete, E., & Navarro, B. (2010). TIPSEM (English and Spanish): Evaluating CRFs and semantic roles in TempEval-2.
4. Kolomiyets, O., & Moens, M.-F. (2010). KUL: Recognition and normalisation of temporal expressions In *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010 Uppsala, Sweden* (pp. 325–328).
5. Negri, M., & Marseglia, L. (2005). Recognition and normalization of time expressions: ITC-irst at TERN 2004. Technical Report WP3.7, Information Society Technologies, February 2005.
6. Strotgen, J., & Gertz, M. (2010). HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL, Uppsala, Sweden* (pp. 321–324).
7. Poveda, J., Surdeanu, M., & Turmo, J. (2007). A comparison of statistical and rule induction learners for automatic tagging of time expressions in english.
8. Jacimovic, J. (2012). Recognition and Normalization of Temporal Expressions in Serbian Texts: *BCI'12,* September 16–20, 2012, Novi Sad, Serbia.

9. Chang, X., & Manning, C. D. (2012). SUTime: A library for recognizing and normalizing time expressions: Angel. In *Eighth International Conference on Language Resources and Evaluation (LREC)*.
10. Mani, I., Pustejovsky, J., & Sundheim, B. (2004) Introduction to the special issue on temporal information processing. *ACM Transactions on Asian Language Information Processing 3*(1), 1–10.
11. http://www.google.com/experimental/.
12. Li, X., & Croft, W. B. (2003). Time-based language models. In *CIKM '03: Proceedings of the Twelfth International Conference on Information and Knowledge Management, New York, ACM* (pp. 469–475).
13. Qamra, A., Tseng, B., & Chang, E. (2006). Mining blog stories using community-based and temporal clustering. In *Proceeding of 15th ACM International Conference on Information and Knowledge Management, ACM* (pp. 58–67).
14. Allan, J., Gupta, R., & Khandelwal, V. (2001). Temporal summaries of news topics. In *Proceedings of the 24th International ACM SIGIR Conference, ACM* (pp. 10–18).
15. Baeza-Yates, R. A. (2005). Searching the future. In *Proceedings of ACM SIGIR Workshop MF/IR*.
16. Ferragina, P., & Gulli, A. (2005). A personalized search engine based on web-snippet hierarchical clustering. In *14th International Conference on World Wide Web (Special Interest Tracks and Posters)* (pp. 801–810).
17. Jones, R., & Diaz, F. (2007). Temporal profiles of queries. *ACM Transactions. Information System., 25*(3), 14.
18. Schilder, F., & Habel, C. (2001). From temporal expression to temporal information: Semantic tagging of news messages. In *Proceeding of the ACL 2001 Workshop on Temporal and Spatial Information Processing*.