A Regular Expression Based Time Expression Recognizer in Hindi

Nitin Ramrakhiyani
DAIICT, Gandhinagar
nitin.ramrakhiyani@gmail.com

Prasenjit Majumder
DAIICT, Gandhinagar
prasenjit.majumder@gmail.com

ABSTRACT

Temporal annotation of plain text is a useful component of modern information processing tasks. In this work we propose a regular expression based approach to time expression identification and classification in Hindi language. When evaluated with the help of human judges, the approach is shown to have a strict F1-measure of greater than 0.75.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Design, Experimentation

Keywords

Time Tagging, Hindi Time Expression Identification, Hindi Time Tagging, Hindi Temporal Tagging

1. INTRODUCTION

Highlighting and processing of temporal information in text has been a challenging problem for the Information Retrieval and NLP communities. This highlighted temporal information has several beneficial applications like question answering (answering the "when" kind of questions), event identification and ordering, and development of biographical summaries. Several approaches have been developed to achieve the desired temporal annotation in English, Italian, Spanish and German[3]. But there is a dearth of such approaches and systems in Indian Languages. In this work the development of a regular expression based approach for recognition of time expressions in Hindi is described.
The described approach utilizes hand crafted regular expressions for identification of time words in input text. The identification rules are modeled as regular expressions and the input text is subjected to regular expression matching. Hence, we use the words 'regular expressions' and 'rules' interchangeably. Once a regular expression match occurs over a word or a set of words, they are identified as a time expression. Further the time expression class associated with the matched rule is assigned to the identified expression. In this work three time expression classes are considered namely DATE-TIME, PERIOD and FREQUENCY.
For evaluation, 100 news articles from the FIRE 2011 Hindi Corpus[2] are chosen randomly and their tagging is carried out using the developed system. We evaluate the results with help of human judges. The system recognizes a large number of time expressions leading to a F1-measure of greater than 0.75.

2. TASK DEFINITION

The Time Entity Recognition task aims to carry out two basic goals explained as follows.

- *Identification of Time expression in given input text*: To fix a boundary around one or more words which denote(s) a proper time expression, in the given plain input text. So given a document D, $\forall$ words w in D, w should be classified as inside a time expression or outside it.
- *Classifying the identified time expression*: To classify the identified time expression as one of the time expression classes given in Table 1. So given the document D, there should be a mapping I such that $I : t \rightarrow x, \forall$ identified time expressions t in D, where $x \in X$.

Table 1: Time Expression Classes

Class(Notation)	Meaning
PERIOD(P)	A time period or duration expression
DATE-TIME(D)	A date or time expression
FREQUENCY(F)	A frequency expression

2.1 Tag Details and Examples

To carry out the delimitation of a time expression and for assigning it an attribute(target classification), tagging the required set of words within an XML tag is an appropriate method. For the purpose of the current experiment we use a single XML tag <TIMEX> to delimit the set of words forming a time expression. This tag is devised to have a single attribute namely TYPE. This attribute holds a value denoting the class of the enclosed time expression namely P, D or F for PERIOD, DATE-TIME OR FREQUENCY expression respectively.

The following lucid illustrations depict a brief representation of the desired tagging.

- Plain Input: पिछले एक साल में युवरज ने छह मैच खेले हैं। (pichle ek saal mein yuvraj ne chah match khele hain.)
  TER Output: <TIMEX TYPE="P">पिछले एक साल </TIMEX> में युवरज ने छह मैच खेले हैं।

- Plain Input: कॉलेज में १४ जून २०१० को परिणाम आयेगा। (college mein 14 June 2010 ko parinam aayega.)
  TER Output: कॉलेज में <TIMEX TYPE="D">१४ जून २०१०</TIMEX> को परिणाम आयेगा।

- Plain Input: हमारा मुनाफा सालाना छह से सात प्रतिशत बढ़ता है। (hamara munafa salana chah se saat pratishat badta hai.)
  TER Output: हमारा मुनाफा <TIMEX TYPE="F"> सालाना</TIMEX> छह से सात प्रतिशत बढ़ता है।

3. THE RULE BASED APPROACH

A collection of words of length one or more, can represent a time expression in many ways. The constituent words can be permuted or replaced by synonyms to form a time expression. Capturing this formation of time expressions can be helpful in identifying time expressions and the rules developed in this approach are aimed at gathering this information. In this section, details about the basic approach are explained.

3.1 Approach Detail

An initial set of rules is developed consisting of as many as possible pivot words (for example दशक (dashak), वर्ष (varsh), सोमवार (somvar), मंगलवार (mangalvar), जनवरी (janvari), फरवरी (farvari), ...). Quantifiers (for example कुछ (kuch), थोड़े (thode), बहुत (bahut), ...) and direction words(for example अगले (agle), पिछले (pichle), नये (naye), ...) are also added to above set to capture different time expression forms. Also a class value(single character) is associated with each rule, denoting the class of time expressions it aims to find. Once the development of this initial rule base is complete, each of the expressions in the initial rule base are searched through the input plain text and if found the matched collection of words are placed under <TIMEX> tags signifying a time expression. The TYPE attribute of the tag is set to the class value of the matched rule.
The rule base is enhanced by an manual iterative process of rule addition. Firstly, the initial rule base was tested on a set of 100 plain files from the FIRE 2011 Hindi Corpus[2]. Based on manual observation of the missing time expressions, rules to identify them were added manually to the existing rule base. While iteratively adding or modifying the rules it is taken care that no conflicts arise within the various new and old rules.

4. EVALUATION

4.1 Evaluation Methodology

After the iterative enhancement of the rule base, evaluation is carried out to gauge its efficiency. Two sets of 50 files each are randomly chosen from the FIRE 2011 Hindi corpus. It is ensured that these sets of files for evaluation are mutually exclusive from the set of 100 files used to develop the rule base. The developed system is then used to tag the evaluation files. The output files are then checked by human judges to determine the correct, partially correct, missing and false positive entries. An annotation is correct if the boundary of the expression and the type assigned both are correct; whereas if the type assigned is correct but the boundary is not exact, the annotation is considered to be partially correct. The manual evaluation was carried out using the GATE Tool[1] and for computation of the recall, precision and F1-measure, a strict mode evaluation was considered under which the partially correct entries are considered as incorrect.

4.2 Experimental Results and Analysis

Tables 2 and 3 highlight the obtained experimental results for both the evaluation runs. In the first run the developed system is able to identify 308 expressions as time expressions against the 295 actually present with an F1-measure of 0.78. The second run results in identification of 371 expressions as time expressions against the actual 351 with an F1-measure of 0.76.

Table 2: Strict Evaluation Results

	Fold 1	Fold 2
Actual	295	351
Total Tagged	308	371
Correct	236	275
Partially Correct	23	16
Missing	36	60
False Positive	49	80
Recall	0.8	0.78
Precision	0.77	0.74
F1-Measure	0.78	0.76

Table 3: Most used Regular Expressions

(Pivot Words)(\s(के(ke)|से(se))\s(Direction Words))? Examples: वर्ष(varsh; year), दिन(din; day)
(Hindi Numerals)+(\s(के(ke)))?\s (Pivot Words)(\s((की(ki)|का(ka))\s)? (अवधि(avadhi)|समय(samay)|भर(bhar)))? Examples: १० वर्ष(10 varsh; 10 years), बीस दिन(bees din; 20 days)

5. REFERENCES

[1] H. Cunningham. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254, 2002.

[2] P. Majumder. Forum for information retrieval evaluation 2011. `http://www.isical.ac.in/~fire/2011/slides/fire.2011.majumder.prasenjit.pdf`, 2011.

[3] I. Mani, G. Wilson, B. Sundheim, and L. Ferro. A multilingual approach to annotating and extracting temporal information. In *Proceedings of the workshop on Temporal and spatial information processing-Volume 13*, page 12. Association for Computational Linguistics, 2001.