# Detection and Normalisation of the Temporal Expression in Hindi Text

Charvee

*Research Scholar in Computer Science*
*HCTM Kaithal (Haryana) – India*

***Abstract*** *- Temporal expressions are those expressions which convey some kind of temporal information i.e. related to time. These expressions can indicate a point in time such as "tomorrow 12 p.m." or a period of time, e.g. "for first 7 months". The task of recognizing temporal expressions from a chunk of text detects the temporal expressions and interprets them. Hence, it essentially consists of two sub tasks of detecting the temporal expressions and normalizing(interpreting) the temporal expressions. Interpretation of the temporal expressions is done in order to make them understandable to the computer algorithms. For Hindi language, the task of recognition has been achieved to some level but the research work related to the interpretation of the detected temporal expression is still in progress. The proposed work attempts to achieve both detection and normalization of temporal expression in texts written in Hindi language with approximately 78% accuracy. Both recognition and normalization make extensive use of the rule-based approach for the detection and interpretation tasks of the temporal entities in the text from news paper articles.*

**Keywords** - *Temporal Expressions, Java-XML Binding, Natural Language Processing*

## 1. INTRODUCTION TO NLP

Natural Language Processing is a field which deals with the interactions between natural languages and computer. This area of Artificial Intelligence is concerned with enabling the machines to be able to understand and communicate with human beings in languages which are natural to them such as English, Hindi,etc. Natural Language Processing is a technique where machine can become more human, hence reducing the distance between human being and the machine. Therefore in simple sense NLP makes human to communicate with the machine easily. There are many applications developed in past few decades in NLP. Most of these are very useful in everyday life for example a machine that takes instructions by voice(e.g. Siri in Apple iPhone). There are lots of research groups working on this topic to develop more practical and useful systems.

Some of the NLP tasks are as follows:
1. Automatic summarization:

Automatic summarization is the process of obtaining the summary or the extracting "important" things from a paragraph. This application is used by the search engines such as Google where the focus is not on the entire paragraph but the main points. There are two ways in which summarization can be obtained, extraction and abstraction. Extraction deals with pulling out the important things from the paragraph, while abstraction uses natural language generation techniques to put out a summary of the paragraph that resembles the summary, a human reader would make off..

2. Information extraction (IE):
Information Extraction is the process of deriving structured factual information which is understandable by the computer algorithms from unstructured text written in a natural language. It is extremely important that the search engines provide the most suitable problem to queries in the least possible time. In order to achieve this, the information should be stored in such a way that it is understandable by the computer algorithms.

3. Machine Translation:
This is one of the most important applications of Natural Language Processing. Translation of a sentence from one language to another, retaining the meaning, is a difficult task. A lot of research has been done on this now in different parts of the world. On a basic level, MT performs simple substitution of words in one language for words in another, but that alone usually cannot produce a good translation of a text because recognition of whole phrases and their closest counterparts in the target language is needed. Solving this problem with corpus and statistical techniques is a rapidly growing field that is leading to better translations, handling differences in linguistic typology, translation of idioms, and the isolation of anomalies.[1]

4. Speech Recognition
Speech recognition (SR) is the inter-disciplinary sub-field of computational linguistics which incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields to develop methodologies and technologies that enables the recognition and translation of spoken language into text by computers and computerized devices such as those categorized as smart technologies and robotics.

It is also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT).

5. Sentiment analysis
Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Sentiment analysis is widely applied to reviews and social media for a variety of applications, ranging from marketing to customer service.

## 2. LITERATURE REVIEW

Time can be represented in two ways- as a point in time or as a period of time. These are the primitive individuals or atoms of time, as Galton [6] calls them- which can be used as the foundations of complex systems for representing and reasoning about time. We note that a variety of terminology is used in the literature when referring to these types of temporal entities. An instant is also referred to as a point in time, a time point, a point or a moment. Van Benthem [7] prefers to use the term period instead of interval, since for him the term `interval' refers to `what lies between boundaries'; this is what others, for example Allen [7], call duration. We will consider the terms `period' and `interval' interchangeable, and treat duration as a feature of a period. Instants differ from intervals by having no duration, which in consequence means that two instants cannot overlap or be contained in one another. In other words, two instant situations happening at the same time in fact happen in the same instant. Intervals have duration, and therefore we can say they have some internal structure (i.e. we can identify subintervals) and distinguish many different relations between intervals. There has been an ongoing dispute as to which entity type is more appropriate as the primary notion for a theory of time. Van Benthem [7] claims that instant based representations are counter-intuitive for modeling time, which, in his view, is a continuum. He claims that we cannot experience a point situation in every-day life (a point in time is an abstract notion and it has no duration); further, human languages do not provide any expressions that refer to points. Therefore, in his view, interval based theories are more natural and better-suited to describing the world. This is not an idiosyncratic view; it has also been expressed by a significant number of researchers in the field of linguistic sciences, perhaps best represented by the well known work of [7].

In discussions as to whether an interval-based approach is more suitable than a point-based theory, [8] does not exclude the possibility of having an approach based on both types of temporal entities. In this vein, [9] developed a computer system for reasoning about time which was primarily based on the logic of intervals, but was extended with new primitive relations and new composition rules over these primitives so that it also covered the logic of point objects. Also the more recent work carried out for the purposes of representing information on the semantic web and processing natural language, and which resulted in the OWL-Time ontology [7], uses notions of both time instants and intervals.

## 3. TEMPORAL EXPRESSIONS STANDARDS

Before discussing what counts as a temporal expression and what not it is important to understand the concept of granularity of temporal expressions. The change in the level of precision can be considered as a change in the granularity of the expression [35]. In order to capture the idea of granularity more in the case of temporal expression, the following example can be considered.

a. I was born in "*1989"*.
b. I was born in "December 1989".
c. I was born on "16th December 1989".
d. I was born at "8:05 pm on 16th December 1989".

The above examples show the various ways in which temporal expressions can be represented. Adding more detail to the temporal expression makes it even a small granule and hence decreasing the granularity. It can said that temporal expression indicating only the year (e.g. 3.1.a in the above case) is coarser expressions than the one indicating the time of event (e.g. 3.1.d ).

**What to consider as a temporal entity?**

Temporal entities can be distinguished into types based on the way that these expressions are referred over a timeline.

1. Point- referring expressions

These are those expressions which represent a point on a timeline and have, thus zero duration. Following are some examples of point referring expressions.

a. Thomas Edison invented the electric light bulb in "1879".
b. India got independence on "15$^{th}$ August, 1947".
c. The incident occurred on "Friday".

Example 3.2 Examples of point-referring expressions

These expressions refer a point on a specific type of timeline and not just of zero duration. Therefore there is nothing wrong in considering, in appropriate circumstances, a century to be a point, although intuitively we think of a century as a period of time stretching over one hundred years. Further distinguishing is done between the date and time type of point expressions.

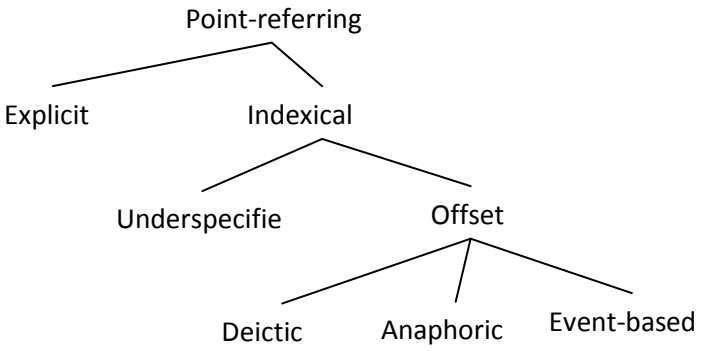


***Fig. 1 Types of point-referring temporal expressions[35]***

Explicit expressions are those expressions which indicate the temporal entity completely and there is no requirement of the domain knowledge or any anchor in order to normalize the temporal entity. Indexical expressions, on the other hand use some kind of external knowledge that the timex is indexed to. Underspecified expressions are those which do not have an exact location on the timeline. With the help of an explicit expression in the vicinity of these, the exact locations can be determined. Offset expressions are those types which need a reference time and simple addition or subtraction from that reference time, the accurate temporal information can be extracted from these expressions. Deictic expressions are offsets which are naturally linked with the agent making the utterance and her temporal locus. Anaphoric expressions are offsets which need to be interpreted with respect to another expression occurring in the text. Event-based expressions identify a temporal entity by means of a reference to an event. Determining the actual point in time referred to by an event-based temporal expression requires identifying the underlying event, determining its temporal location, and then calculating the offset.

a. Attacks on Indian Parliament were carried out on "*13th December, 2001*".
b. They got married in "*January*".
c. Let's plan the reunion "*this weekend*".
d. I am going to the Himalayas "*next week*".
e. Earthquake was first felt at 7:40 pm and again "*half hour later*".
f. "*Ten seconds after*" the second explosion the plane hit the ground.

Examples of explicit(a), underspecified(b), deictic(c), anaphoric(d) and event-type(e) temporal expressions

2. Period referring expressions

These are references which indicate how long something (for example, a process or event) lasted, lasts or will last. Here are some examples:

a. The accounts are paid in full for the "six months ended March 31".
b. The war of Mahabharata lasted "18days".
c. The ship was at sea for "two weeks".

Example 3.4 Examples of period-referring expressions

These types of entities are called duration type entities as per the TIMEX standards and their value is denoted with a prefix "P" indicating a period. There are some other kinds of period expressions which denote some kind of recurrence such as:

a. "Every year", authorities raise the admission fees in the schools.
b. I go swimming "every four days".
c. The PM goes on a foreign trip "every month".

Example 3.4 Examples of re-occurring expressions

Such expressions are called as set expressions in TIMEX standards.

Some temporal expressions may be non-specific, in the sense that they refer to temporal entities that cannot be placed on a timeline. We distinguish two types of nonspecific city: generic and indefinite. In the first case, the expression is typically part of a generally-holding statement or belief, or a description of a universal law or state; e.g. February is the shortest of all the months. In the second case, the expression refers to an entity which is linked to some eventuality that takes place in a specified time but its temporal location cannot or is not supposed to be determined, e.g. I was born on a Sunday morning.

## 4. PROPOSED WORK

The current work is the development of the software system which is able to parse the text written in Hindi and recognize the temporal expressions in the text and is able to normalize these expressions, i.e. represent their value in such a way that is recognizable by the computer algorithms. For Hindi language detecting the temporal expressions has been done using rule-based and machine learning methods by [40] but the normalization part has not been carried out till now. *This work aims at developing the system which can not only detect but also normalize the value of temporal expression.*

**APPROACH USED**

The current work is based upon rule-based approach for the purpose of detection of temporal expressions in the text written in Hindi language. It has been argued that for the purpose of normalization, there is no alternative to the rule based approach.[41]. Thus, for both detection and normalization of temporal expressions rule-based approach is used. In order to develop rules, Java Regular Expression (Regex) matching system[42]. Each rule is implemented as a regular expression. This eases the process of handling a large number of rules and also makes the identification process effective. For example, for separate rules like १८-०१-२०१४ (18-01-2014), १८/०१/२०१४ (18/01/2014), १८.०१.२०१४ (18.01.2014), the single regular expression, (([०१२३][०१२३४५६७८९])[-/.]([०१][०१२३४५६७८९])[-/.]([०१२३४५६७८९]{4})) is included. English representation of this expression is (([0123][0123456789])[-/.] ([01][0123456789])[-

/.]([0123456789]{4})). Similarly for separate rules like अगले दिन (agale din), अगली रात(agale raat), पिछले दिन(pichle din), पिछली रात(pichle raat), and expressions like that, instead of including them as they are, they are obtained by performing a Cartesian product of sets (अगले,पिछले,…) and (दिन,रात,…). Thus the total effort spent in the writing separate rules for each of these cases is minimized. Each of the rules regular expressions in the developed rule base is applied to the input text through regular expression matching. Once a match is found, it is placed under <ILTIMEX>and </ILTIMEX> tags signifying a time expression.

4.2 System architecture

Unlike rule based taggers in English and other languages, which require some form of linguistic preprocessing like POS tagging and parse information of the input text, this approach does not require any form of linguistic preprocessing and hence no such preprocessing is carried out. Certain undesired text formations like multiple spaces between words and unknown characters were harmlessly removed from the input text to ease rule application. The figure below represents the position of Recogniser and Normaliser of the system. The Recogniser works on the chunk of text while the Normaliser has the matched text to operate on. Along with the matched the matched text, Recogniser returns the regular expressions' type it has matched the expression with. For example, in the case of matching expressions such as "24 मई" from the text, Recogniser extracts "24 मई" and also tells the Normaliser that it is date type of expression which makes the normalization of the expression easier.

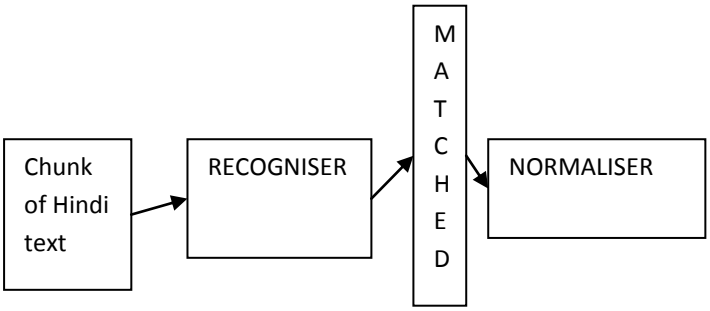


***Fig.2 Broad view of the system***

Since Hindi language does not have capitalization ( i.e. first letter capital in case of Nouns or names of Nouns), there is no requirement of preprocessing the text and the chunk is directly fed to the Recognizer.

The Recognizer

Recognizer receives from the text, ( | ) delimited sentences which in Hindi language is considered as a full-stop. Since the target corpus consists of news articles only hence the delimiter is chosen as full stop only and question-mark(?) is not considered.(Although some sentences in other literary works may end in a question mark, but news articles are only informative in nature, hence question-mark is not taken as a delimiter). There are about 30 rules at present in the rule-base which combine based on the Cartesian product described above and hence make the rule-base more efficient. However since the corpus used for the present work consists of news article only hence the rules are written keeping that in mind. This implies that the rules written might not perform well in case of other sources such as literary works, etc. Recogniser runs the entire chunk of text against all the regular expressions written for Date, Time, Duration and Set type of expressions. Once a match is found in the text, the matched text is picked from the text along with the type of regular expression with which it was matched. The contents of the chunk till the matching text are placed intact in the output while the matching text and the regular expression with which it matched are sent to the normaliser so that TIMEX tags can be surrounded to it. There may be possibility that there are more than one temporal expression in the same sentence. In order to counter that, once a match is found in a sentence, the sentence is split into two halves. One half of the sentence contains the part of the sentence from the beginning to the point where the matching text was found. This half has no more temporal expression gone undetected. Thus it needs to be present in the output as such and hence it is added to output. The other half contains the contents of the original sentence from the point where the matching text was found till the last word and may contain the temporal expressions that have not yet been seen. Hence the other half is again sent for detection to the recognizer where it is checked for any match again. As per the TIMEX2 standards, the <TIMEX> tag should have an identifier which should be unique in the document and should be a monotonically increasing number with an increment of one. Recogniser gives the identifier to the temporal expression and then sends the matched text and the regular expression with which it matched to the Normaliser. Normaliser is responsible for assigning the type, i.e. whether a Date, Time, Duration or a Set type of temporal expression has been found. In addition, Normaliser interprets the value of the matching text and hence is responsible for assigning the value to the <TIMEX> tag.

The recognizer was designed in such as way that in addition to just detecting the temporal entities present in the chunk of text, it also helps the normaliser in interpreting its value. Since the implementation is done in Java language, hence rules for matching the written as Regular expressions (regex) patterns using the Regex package[42]. These regex patterns are placed inside a Linked HashMap which has regex pattern as the key and their type is saved as the corresponding value in the HashMap. When there is a match with the regex pattern, the corresponding type

is also picked as the value pointed by the key. Both matching text and value(type of the regex pattern which matched) are sent to the normaliser.

## 5. RESULTS AND CONCLUSION

The accuracy of the system is measured by the precision and recall values. Precision and recall is calculated using the manually tagged documents as the gold standard. By drawing the comparison between the gold standard document and the proposed system generated document, these values can be reached to.

### GATE TOOL

For the measuring the accuracy of the proposed system, GATE(General Architecture for Text Engineering) developer tool[44] was used. A module of this tool is Annotation diff tool which is used to differentiate between the annotations (or tags) present in two XML documents. This tool identifies the tags present in the key document i.e. the gold standard document, and compares these tags with that of the response document, i.e. the system tagged document. Based on the presence or absence of tags in the key and response documents, the Annotation diff tool calculates the values of four parameters, vis-à-vis true positives, false positives, true negatives and false negatives. Using these parameters the precision and recall values of the particular document is generated. Hence, the Annotation tool diff tool supplies the four values after comparing key and response documents,:

1. Correct
2. Partially correct
3. Missing
4. False positives

ILTIMEX tag, along with its attributes and value, in key document is compared with that of the response document. Correct states the number of tags in the key document that have been the exact match with the tags of response document. In case there is some difference in the attributes or value of the tag, the Annotation diff tool places such cases with the partially correct category. Thus, partially correct gives the number of tags that have been partially correct. Missing states the tags in the key document that are not present in the response document. These are the false negative cases. False positives are those tags which have been incorrectly identified by the proposed system as a tag. These tags are not present in the key document.

### RESULTS

Annotation diff tool calculates three types of precision, recall and F-measure value for the particular document: Strict, Lenient and Average. Strict values are calculated considering the partially correct as the missed cases. While the Lenient takes these values as the correct cases. Average case takes up the arithmetic mean of the Strict and Lenient values. While calculating the accuracy of the proposed system the average values have been taken up. The figure below shows the annotation diff tool output for corpus document, C-23.xml. The missed tags have been given a pink colored background while for false positives have orange colored background. In this figure, the system could not detect the presence of time related information and the tag was missed. On the other hand, there was a tag that was incorrectly considered as a temporal entity by the system and it has been judged as the false positive. The overall (or average) F-measure comes out to be 0.83 on the scale of 1, which is 83%.

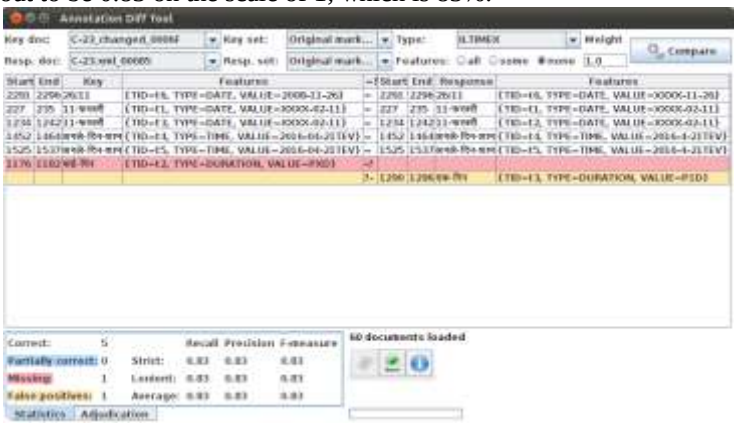

**Fig. 3 Annotation diff tool for C-23.xml**

Considering output of the Annotation diff tool of another document,C-21.xml in the figure below, all the tags in the key document have been correctly identified in the response document. So in such a case the F-measure for the particular document comes out to be 1.00 on the scale of 1.00, i.e. 100% for this particular document.

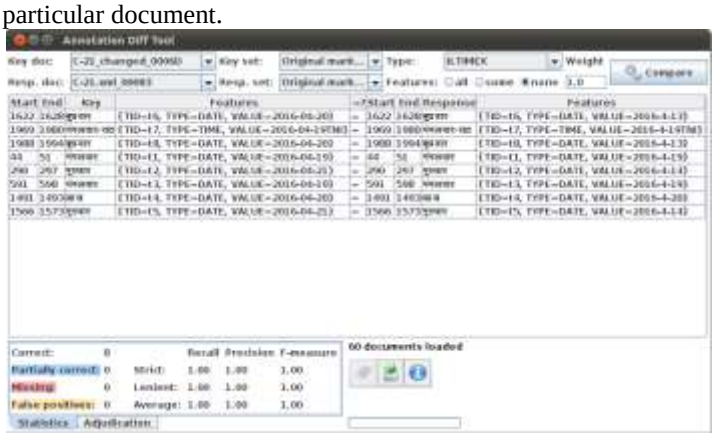

**Fig. 4 Annotation diff tool for C-21.xml**

In this manner, the differences between the annotations of both the documents are captured. The arithmetic mean of average F-measure of all the documents is taken and the accuracy of the proposed system hence comes out to be approximately 78%.

### CONCLUSION

Currently the rule base of the proposed system consists of about 30 rules and is able to produce approximately 78% accuracy. Expanding the rule base hence is expected to improve the accuracy of the system. The expansion of rule-base for the purpose

of detection implies a similar expansion in the rules for normalization, since it also relies heavily on rules. Also, the proposed work deals with the news articles only. Therefore, it can be seen as a shortcoming of this system that it may not be able to repeat these results for the other Hindi texts such as novels, etc.

The proposed work relies on the development of the rule-base. Statistical machine learning methods are not being used in the development of the application. So, as a future extension of the proposed work, using features such as 'contains Digit', 'IsNoun' , etc the statistical classifiers based on Machine learning methods such as Conditional Random Field, (CRF), Hidden Markov Model(HMM) can be used. A comparison can be drawn between the performance of rule-based approach and the statistical methods approach.

It may also be noted that this proposed work deals with the normalization of the detected Hindi temporal expressions which has not been done prior to this. Hence it lays the foundation of such works in the future. Hindi is the national language of India but it does not get its due importance. After this proposed system comes across the eyes of the researchers of the NLP community, the interest and motivation for developing the NLP applications is sure to increase.

The result of the proposed work has lead to the development of the NLP application in Java which detects and normalizes the temporal expressions in the Hindi language with an accuracy of 78% on the corpus containing news articles.

## REFERENCES


[1] *Albat, Thomas Fritz*. "Systems and Methods for Automatically Estimating a Translation Time." US Patent 0185235, 19 July 2012.

[2] *Bar-Hillel, Yeheshua* "A demonstration of the nonfeasibility of fully automatic high quality machine translation", Language and Information: Selected essays on their theory and application (Jerusalem Academic Press, 1964), pp. 174–179.

[3] *Madsen, Mathias*, "The Limits of Machine Translation (2010)". docs.google.com

[4] Speaker Independent Connected Speech Recognition- Fifth Generation Computer Corporation. Fifthgen.com. Retrieved 2013-06-15.

[5] *Reynolds, Douglas; Rose, Richard).* "Robust text-independent speaker identification using Gaussian mixture speaker models" , *IEEE Transactions on Speech and Audio Processing (IEEE) **3** (1):72–83.doi:10.1109/89.365379.* ISSN 1063-676. OCLC 26108901.

[6] *Huttunen, S., Yangarber, R., Grishman, R*, "Complexity of event structure in information extraction". In: Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002). Taipei (2002)

[7] *Jakub Piskorski and Roman Yangarber,"* Information Extraction: Past, Present and Future", Multi-source, Multilingual Information Extraction and Summarization 11, Theory and Applications of Natural Language Processing, pp. 23-49 DOI 10.1007/978-3-642-28569-1__2, © Springer-Verlag Berlin Heidelberg 2013,

[8] *Andersen, P., Hayes, P., Huettner, A., Schmandt, L., Nirenburg, I., Weinstein*, S."Automatic extraction of facts from press releases to generate news stories". In: Proceedings of the 3rd Conference on Applied Natural Language Processing, ANLC '92, Trento, pp. 170–177.Association for Computational Linguistics, Stroudsburg (1992)

[9] *Riloff, E.*"Automatically constructing a dictionary for information extraction tasks". In: Proceedings of Eleventh National Conference on Artificial Intelligence (AAAI-93), Washington, DC, pp. 811–816. AAAI/MIT (1993)

[10] *Phillips,W., Riloff, E.*,"Exploiting strong syntactic heuristics and co-training to learn semantic lexicons". In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002) (2002)

[11] *Chinchor, N.*, "MUC-7 Named Entity Task Definition, version 3.5, 17", Proceedings of the Seventh Message Understanding Conference (MUC- 7), Morgan Kaufmann Publishers, September 1997.

[12] *Grishman,R.*, " The NYU System for MUC-6 or Where's the Syntax?", In Proc. Sixth Message Understanding Conference (MUC-6), Columbia, MD, November 1995.

[13] *Iwanska, L., Croll, M., Yoon, T., and Adams, M.*, "Wayne state university: Description of the UNO Natural Language Processing System as used for MUC-6", In Proc. Sixth Message Understanding Conference (MUC-6), Columbia, Morgan-Kaufmann Publishers, 1995.

[14] *Greenwood, M. A. and Gaizauskas, R.*, "Using a Named Entity Tagger to Generalize Surface Matching Text Patterns for Question Answering", In EACL03: 10th Conference of the European Chapter of the Association for Computational Linguistics, Budapest, Hungary, 2003.

[15] *Toral, A., Llopis, F., Munoz, R., and Noguera, E.*, "Reducing Question Answering Input Data using Named Entity Recognition", In Proc. 8th International Conference on Text, Speech & Dialogue, 2005.

[16] *Molla, D., Zaanen, M., and Smith, D.*, "Named Entity Recognition for Question Answering", In Proc. ALTW 2006

[17] *Babych, B., Hartley, A., and Atwell, E.*,"Statistical Modelling of MT output corpora for Information Extraction", In Proc. Corpus Linguistics conference, Lancaster University (UK), pp. 62-70, 28 - 31 March 2003.

[18] *Tsai, R. T. H., Sung, C. H., Dai H. J., Hung, H. C., Sung,T. Y., and Hsu, W. L.*, "NERBio: Using Selected Word Conjunction, Term Normalization, and Global Patterns to Improve Biomedical Named Entity Recognition", BMC Bioinformatics, 7(Suppl 5):S11, 2006.

[19] *Zhou, G., Zhang, J., Su, J., Shen, D., and Tan, C.*, "Recognizing Names in Biomedical Texts: a Machine Learning Approach", Bioinformatics, vol. 20, no. 7, pp. 1178-1190, 2004.

[20] *Sikdar, U. K., Ekbal, A., Saha, S.*, "Modified Differential Evolution for Biomedical Name Recognizer", Computational Linguistics and Intelligent Text Processing, pp. 225-236, 2014.

[21] *Lev Ratinov, Dan Roth,* "Design Challenges and Misconceptions in Named Entity Recognition", In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), Association for Computational Linguistics, pp. 147–155, Boulder, Colorado, June 2009.

[22] David Nadeau, Santoshi Sekine,"A survey of named entity recognition and classification", National Research Council, Canada/New York University, 2007

[23] *Coates-Stephens, Sam.* "The Analysis and Acquisition of Proper Names for the Understanding of Free Text" . *Computers and the Humanities* 26.441-456, San Francisco: Morgan Kaufmann Publishers. 1992

[24] *Fleischman, Michael.* "Automated Subcategorization of Named Entities" . In Proc. Conference of the European Chapter of Association for Computational Linguistic, 2001

[25] *Cucerzan, Silviu; Yarowsky, D.*"Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence". In Proc. Joint Sigdat Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.1999.

[26] *May, Jonathan; Brunstein, A.; Natarajan, P.; Weischedel, R. M.* "Surprise! What's in a Cebuano or Hindi Name?" *ACM Transactions on Asian Language Information Processing* 2:3.pp.169-180, New York: ACM Press, 2003.

[27] *McCallum, Andrew; Li, W.”*Early Results for Named Entity Recognition with Conditional Random Fields, Features Induction and Web-Enhanced Lexicons”. In *Proc. Conference on Computational Natural Language Learning*. 2003

[28] *Ekbal, A. and Saha S.”*Weighted vote-based classifier ensemble for named entity recognition: A genetic algorithm-based approach”. *ACM Trans. Asian Lang. Info. Process. 10*, 2 , 2011.

[29] *Ferro, Lisa; Gerber, L.; Mani, I.; Sundheim, B.; Wilson G.* “TIDES 2005 Standard for the Annotation of Temporal Expressions”. The MITRE Corporation.2005

[30] *J. Hoffart, F.M. Suchanek, K. Berberich, and G.Weikum*. “YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia”. Artificial Intelligence 194, pp28–61, 2013.

[31] *M. Matthews, P. Tolchinsky, R. Blanco, J. Atserias, P. Mika, and H. Zaragoza*. “Searching through time in the *New York Time”*.. In *Proceedings of the HCIR’10 Workshop*. pp.41-44 . 2010

[32] *J. B. Michel, Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, S. Pinker, M. A. Nowak, and E. L. Aiden*. “Quantitative analysis of culture using millions of digitized books”. Science 331, 6014, pp. 176–182. 2011

[33] *B. Kahle*. “Preserving the internet”. *Scientific American Magazine* 276, 3, pp. 72–73. 1997

[34] *A. Galton.* “Time and change for AI”. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, Handbook of Logic in Artificial Intelligence and Logic Program-ming, volume IV, pp. 175-240. Oxford University Press.

[35] *Mazur Pawel*, “Broad-Coverage Rule-Based Processing of Temporal Expressions” PhD Thesis, Macquarie University, Centre for Language Technology, 2012

[36] *I. Mani and G. Wilson*.”Robust temporal processing of news” In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL), pp. 69-76, Hong Kong, October. Association for Computational Linguistics.2000

[37] *Jannik Strotgen and Michael Gertz*. “Heideltime: High quality rule-based extraction and normalization of temporal expressions”. In Proceedings of the 5th International Workshop on Semantic Evaluation, pp. 321-324. Association for Computational Linguistics, 2010.

[38] *Angel X Chang and Christopher D Manning.* “Sutime: A library for recognizing and normalizing time expressions”. In LREC, pp. 3735-3740, 2012.

[39] *Yu-Kai Lin, Hsinchun Chen, and Randall A Brown*. “Medtime: A temporal information extraction system for clinical narratives”. Journal of biomedical informatics, 46:S20-S28, 2013.

[40] *Ramrakhiyani, N. and Majumder, P*. “Approaches to temporal expression recognition in Hindi”. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 14, 1, Article 2 (January 2015), 22 pages. DOI:http://dx.doi.org/10.1145/2629574, 2015

[41] *Hector Llorens, Leon Derczynski, Robert Gaizauskas, Estela Saquete*. “TIMEN: An Open Temporal Expression Normalisation Resource” , LREC, page 3044-3051. European Language Resources Association (ELRA), 2012

[42] https://docs.oracle.com/javase/7/docs/api/java/util/regex/package-summary.html

[43] www.joda.org/joda-time

[44] https://gate.ac.uk/download

[45] *UzZaman, N., Llorens, H., Derczynski, L., Verhagen, M., Allen, J., and Pustejovsky*, J. “SemEval Task 1: TempEval-3: Evaluating time expressions, events, and temporal relations”. In Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval’13) in conjunction with the 2nd Joint Conference on Lexical and Computational Semantcis (* SEM ’13). Association for Computational Linguistics,June. 2013

[46] *Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Moszkowicz, J., and Pustejovsky, J*. “The TempEval challenge: Identifying temporal relations in text”. *Lang. Resources Eval.* (Special Issue on Computational Semantic Analysis of Language: SemEval-2007 and Beyond) *43*, 2, pp. 161–179.2009

[47] *Verhagen, M., Sauri, R., Caselli, T., and Pustejovsky, J*. “SemEval-2010 task 13: TempEval-2”. In Proceedings of the 5th International Workshop on Semantic Evalua- tion. Association for Computational Linguistics, pp. 57–62.

[48] *Palchowdhury, S., Majumder, P., Pal, D., Bandyopadhyay, A., and Mitra, M*. “Overview of FIRE 2011”. In *Multilingual Information Access in South Asian Languages*, Springer, pp. 1–12. 2013

[49] *Ramrakhiyani, N. and Majumder, P*. “Temporal expression recognition in Hindi”. In *Mining Intelligence and Knowledge Exploration*. Springer, pp 740–750. 2013

[50] Venu Dave and et al.” Sentiment Analysis of Tourists Opinions of Amusement, Historical and Pilgrimage Places: A Machine Learning Approach” in International Journal of Computer Trends and Technology Volume 46 - No 2 April 2017.