International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 10 (2015) pp. 25329-25346

http://www.ripublication.com

Analysis of Temporal Expression In The News Articles Using Temporal Taggers

R. Suganya Devi[1], R. Vidhya[2], Dr. D. Manjula[3]
[1]*Research Scholar, Department of Computer Science and Engineering, suganyawaran2001@gmail.com ,College of Engineering Guindy, Tamil Nadu, India.*
[2]*Post Graduate Student, Department of Computer Science and Engineering vidhu.cs111@gmail.com, College of Engineering Guindy, Tamil Nadu, India.*
[3]*Professor, Department of Computer Science Engineering, manju@annauniv.edu, College of Engineering Guindy, Tamil Nadu, India.*

Abstract

Information retrieval deals with different types of information (e.g., text documents, images, and videos) and different problems (e.g., searching, organizing, and storage). The primary goal of this field has been searching text documents (e.g., web pages and newspaper articles) i.e., in a huge document collections finding those text documents that satisfy the users need. One of the prominent emerging research area in the domain of information retrieval is temporal information retrieval. Exploiting temporal information in documents and queries to improve the effectiveness of information retrieval. In such temporal documents are created and edited over the time, web archives, news archives, blogs and emails are example of temporal documents. The major challenges in such document collections are extracting temporal information and normalizing the temporal information, i.e. all temporal expressions normalized to their standard format. Many temporal taggers are used to extract such information like TarSqi, Heidel Time, SUTime etc. This paper contributes an analysis and per evaluation of the two temporal taggers TarSqi and Heidel Time.

Keywords: Temporal Information Retrieval, Temporal Taggers

Introduction

Information retrieval is the process of retrieving unstructured records i.e. records primarily in free from language text, from a huge collection of documents relevant to an information need. In information retrieval, a query is a request to retrieve information. Information retrieval tries to find and retrieve documents that are relevant to the queries which are usually generated by the user. A document is said to

be relevant if it satisfies the users need. Since the document collection is voluminous, a large number of documents may be termed to be relevant to the user. Emerging field in the information retrieval is temporal information retrieval, in which identification and extraction of the temporal information is an important processing. The web contains large collections of data and so a huge amount of temporal information available in the web. It is essential to identify the temporal information from different no of web sources such as news archives, twitter, and personal emails. Web documents that contains temporal information in different places such as metadata, content techniques and extracting temporal information from the query. When the time dimension can be incorporated into the search, it will increase the retrieval effectiveness. Even though the publication time and creation time is available in the documents, it is difficult to identify the correctness and trustworthiness of a document. At first, the metadata time of documents that are preserved in the past might not always be readable and interpretable in present times. Second, an accurate and trustworthy timestamp for a web document is difficult to find because of the decentralized nature of the web, where the document can be relocated and its time metadata made unreliable. Moreover, in a web warehouse or web archive there is no guarantee that a document's creation date and the time of retrieval by the crawler are related. In this paper, the documents contents and the time of the topic of documents' contents is analyzed. Extraction of information done using the temporal taggers that was developed by Marc Verhagen [1] et.al and Jannik Strotgen [2] et.al. TarSqi toolkit automatically identifies temporal information and events from the natural language texts. Heidel Time is also used for extracting and normalizing the temporal information from the documents. It is based on the regular expression pattern to extract the temporal information. In this paper, temporal expressions reside on the content is analyzed. Using the temporal taggers the temporal expressions are extracted and performance of the temporal taggers is evaluated.

Temporal Annotations of Documents

The following sections describe the various types of temporal dimensions available in a document. A simplified definition of concepts related to time and the relationship between time and events are described. A graphical representation of timeline is described to better understand the passage of time. Then the different types of temporal expressions and the methodologies included in the extraction of temporal expressions are described.

Concepts Related To Time:

Bruce [3], represented the temporal references formally. He described time as an ordered pair, (time, ≤), here the time is described as a set that contains several elements. The elements are called time points and ≤ described a relation between times. Allen [4], described another formal approach it contains a set of thirteen possible temporal relationship between any two time intervals. A particular moment or point in time value is called as instant of time. Point in time values are defined by units. The range of the unit varying from finest granularity to the coarsest granularity

such as day, week, month, semester, quarter, year, decade, and century. The day can also be split into hours, minutes, seconds, fractions of a second, and so. In the calendar, the time values are physically represented as several granularities. Nowadays the most widely used calendar is the Gregorian calendar. It follows the ISO-8601:2004 standard format. In this calendar the date is usually represented as YYYY-MM-DD, here [YYYY] represents as a year, [MM] represents as a month, and [DD] represents as a day. Also some specialized calendars also available, such as sports, fiscal, business calendars.

In the database research areas, two types of time are identified by [Snodgrass and Ahn 1985]. They are valid time and transaction time. The events occur at specific period of time and that time is usually described as the valid time. The events or facts are stored into the database at the specific period time is called transaction time. In the web page, time can be represented implicitly and can be referred from the content of the web pages. This time is referred as focus time. The focus time can be described as set of time intervals rather than as a single point in time, valid time is an example. The time that the web page is created is called creation time, at the same time it can be modified or published at another time interval. These time intervals are referred as transaction time. Also the birth time and end time is considered. Birth time is the first crawling date of the document and end time is the most recent crawling date of the time.

The other types of time in the documents that are also considered are reading time and document age. The reading time considers as a user's reading the documents after performing the search. The difference between the reading time and the timestamp is considered as the document's age.

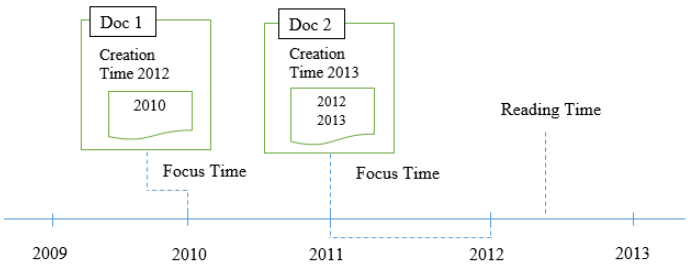

Figure 1: Timeline

Fig 1 shows the documents showing various creation dates, focus dates and reading times etc.

A. Events And Time

Time is naturally incorporated with the events described by the Setzer and Gaizauskas [6]. For example "I went to Africa on Monday" In this phrase Monday is the temporal expression that defines "*went to Africa"*. Events can be described as a happening at a determinable time and place. They may take place with or without the participation of human agents. Also, time can be mapped by the bi-dimensional spatio-temporal view.

Allan et al. [7] , introduced a system to automatically determine and track events through the topic detection and tracking initiative. TDT is the first automatic news management system that identifies the occurrence of the new events using that to track the time of the events. A classical hypothesis test that can be used to discover the temporal features that was proposed by Swan et al. [8]. It is used to identify the important topics in text documents. Using temporal similarity measure of one document can be compared with another and this alternate solution was proposed by the Makkonen et al. [9].Topic events that can be correlated with texts was proposed by Shaparenko et al. [10]. The changes in the events also affected the text used in the documents. K-means clustering algorithm were used to detect the changes in the text. In K-Means algorithm, each clusters were represented as an important topic. The number of documents stored in a cluster, is used to describe the popularity of the document over time.

B. Timelines

Timeline is used to represent the sequence of events also referred as chronology. In which the events are listed by graphical representation within a particular time interval. A timeline is very useful to visualize and comprehend time lapses between events, durations and the also the overlap of spans and events.

Timelines are particularly useful for studying history, as they help us to visualize the change over time. Timelines are generally constructed depending on their purpose. The above mentioned Fig 2 is a timeline of Google Nexus Smartphones.

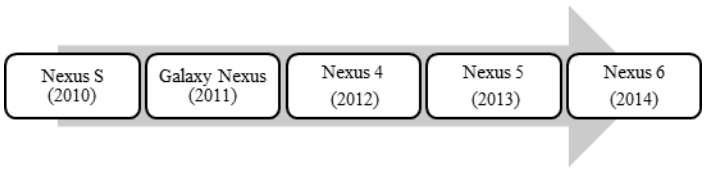

Figure 2: Timeline of Google Nexus

C. Temporal expressions

In the text, the temporal expressions are described as a sequence of tokens such as words,numbers, and characters. These are expressed as a duration or a point in time and frequency. Examples for a point in time, a duration and a frequency: " Anitha was born on <TIMEX> 10 June, 1988</TIMEX>,The show lasted <TIMEX>10 minutes</TIMEX>, The pump circulates the water <TIMEX>every 10 hours</TIMEX>. The temporal expressions can be classified into three types Implicit temporal expressions, Explicit temporal expressions, Relative temporal expressions. Implicit temporal expressions are associated with events in the text, which are implict temporal nature. For example, take the expression "New Year". This expression contains a temporal nature implicitlly. Atleast absolute time expressions are required to establish the accurate temporal values. Setzer and Gaizauskas[11], the precise moment in time can be explicitly specified in the timeline without any further knowledge that can be described as explicit temporal expressions. Explicit expressions depends on the granularity level, for example "12.09.2013" described as a

day's granularity, September 2013 described as a month's granularity, 2013 described as a year's granularity. Relative temporal expressions depend on the creation date or publication date or some other date near in the context to point out time rather than refering to the time explicity. For Example the relative temporal expressions such today, last Monday, or in 5 minutes, before this are temporal expressions relative to the document timestamp i.e, they point out to the actual time indirectly by depending on direct time references. The next section describes as how to extract temporal information from the contents.

D. Extraction of temporal information

For unstructured textual documents, the preprocessing step can include tokenization, part-of-speech tagging, stop word removal, stemming and lemmatization. In the temporal information extraction process is also required preprocessing step for each document because it is a non-trivial task. The preprocessing stage involves five steps: Tokenization, Sentence extraction, POS Tagging, Named entity recognition, and Semantic role labeling. Tokenization splits a document into a list of words or tokens. In English the characters used as sentence delimiters are a period, a question mark, an exclamation mark, or a comma. In each text that contains set of sentences that can be identified by the sentence extraction process.

Tagging is the process of labeling each token with its part-of-speech (POS) in the sentence, such as, a noun, a proper noun, an adjective, a verb, a determiner, or a number. POS tagging helps in removal of irrelevant words (e.g., adjectives, or determiners) and also can be used to reduce ambiguity of words with several meanings (a noun or a verb). Moreover, tagged tokens are useful for the stemming process. In the documents contains nouns, verbs, etc. Using the Named-entity Recognition (NER), the proper nouns are to be identified. The final step of this preprocessing is Semantic Role Labeling; detection of the semantic arguments associated with the predicate or verb a sentence and their classification into their specific roles. At the same time detection of temporal expression start before the NER process. The extraction of temporal expression is an independent task. There are three steps involved in the extraction of temporal information. First step involves recognition or extraction of temporal expressions. The temporal expressions that are found in the first step, are normalized for the TIMEML specifications i.e. the type of temporal expressions are identified. In the TIMEML temporal expressions classified into four types: Date (for calendar date), Time (for time of day), Duration (for spans of time), and Set (for recurring times). This task is known as Normalization of the temporal expressions. The third step is temporal annotation in which using the standard format like Time ML the temporal expressions are expressed. The figure describes the steps of temporal information extraction process.

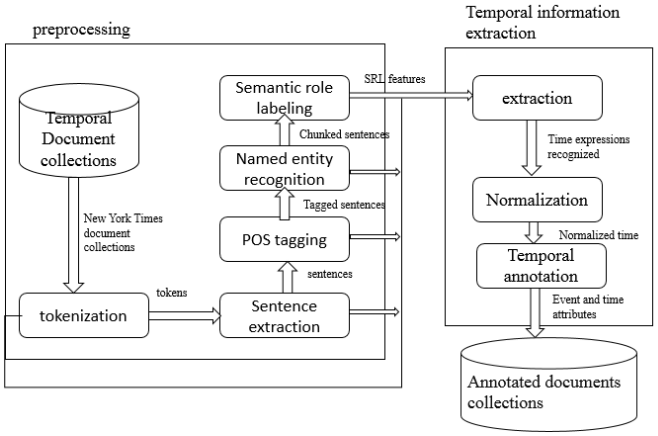

Figure 3: Overall Layout of Temporal Information Extraction

Temporal Information Extracted From The Web Resources

Web contains large no of data to obtain the relevant data. So the temporal information available in the documents is very high. This information stored into the different web sources, from social networks such as twitter to web query logs. The temporal features extracted from various web sources and three different approaches that are related to extracting such information are described below. The temporal information reside in the metadata, content of the documents and, web query logs.

Metadata

The document's metadata contains the time information can be extracted from this approach. Example of the metadata information are the publication time, creation time of the documents and, the last modified date of the documents. The URL of the documents or the documents contain the anchor texts connected to the links that pointing to the documents also include the temporal information that can be extracted in this approch. News collections mostly contains this information. The metadata temporal information are inapt for solving relative temporal expressions because the publication or creation time differ from the focus time. In case, a doucument with creation time sep 2010 but the content contains the date 12 Nov 2013. This can be resolved by content approaches.

Content Approaches

The webpage contains temporal information in the content. That time is described as documents focus time. In the content approach this temporal information are analyzed and the expressions are extracted. But the difficulty level has increased in the content approaches rather than metadata approaches because the linguistic analysis of text. Also web is multi-lingual, highly multi-domain, multi-cultural, heterogeneous is common. For an example, the term "Independence Day" refers to different dates in different regions of the world. In such a case, the time tagger must be capable of identifying the appropriate time based on different language. Jatowt et al.[12],

described the range and typical granularity of the temporal expressions in the online news articles. In the news articles that contains daily-granularity expressions. This expressions that refered to the immediate past,the present, or the immediate future.

Webquery Logs

The temporal information are extracted from the web query logs. From the query perspective it is classified into two types: query time stamp and query focus time. The query time stamp described the issued date of the query. The server activity is recorded over the time in the web query logs. The recorded information is finally obtained from the web query logs. The second one query focus time is the user's referring the query at different time intervals it is mostly referred as content time of the documents. At the same time the queries are classified into two types: implicit temporal queries and explicit temporal queries. Some temporal expressions that contain a concrete date, easily resolved expressions and indicate a certain time period these are likely to be part of the explicit temporal queries. Nunes et al [13], identified 1.5% of queries are explicitly specified. Eliminating the false positive temporal expressions (e.g., "form 2013" or "college 1998") from this queries this value was reduced to 1.21 % by Campos et al [14]. Implicit temporal queries does not contain temporal expressions explicitly. Jones and Diaz [15], divides such queries as three types: atemporal queries, temporal unambiguous, temporal ambiguous. Some queries are not sensitive to time that are specified as atemporal queries (e.g., "cat"). Temporal unambiguous queries specified some concrete period of time. Finally, periodical or aperiodic events are referred to as a temporal ambiguous queries.

Temporal Taggers

Extraction of temporal information done by temporal taggers. It follows rule based approaches based on linguistic-grammar-based or regular expressions techniques. In the past few years, the most important research area is temporal taggers. And also temporal taggers mostly available for the English language and news archives. Some challenges in this processes are the document creation time, delimiting, classifying and normalizing temporal expressions, recognizing events or delimiting their temporal order [15]. The lack of a comprehensive collection of annotated texts with temporal information in different languages provides an added challenge. Simply finding part of speech functions has a lot of language intricacy and ambiguity.

The temporal taggers such as *TempEx* [Mani and Wilson 2000], *GUTime*, *Annie*, *Heidel Time* [Strötgen and Gertz 2010a], and *SuTime*14 [Chang and Manning 2012] can be used for extracting and normalizing temporal expressions. They are evaluated and measured separately. This identifies temporal expressions and normalizes them to some predetermined standard format. Precision (P), RecallI and F-Measure (F) can be computed as evaluating parameters.

$$\text{Precision} = \frac{tp}{tp + fp} \text{Recall} = \frac{tp}{tp + fn}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}.$$

fp (False positive) is the number of expressions incorrectly identified by the system whereas fn (False Negatives) is the number of expressions incorrectly identified as non-temporal expressions.

One of the very first temporal expression tagger developed was TempEx. It extracts explicit and relative temporal expression based on prewritten rules and labels them with TIMEX2 tags. The process starts with tokenization of words and sentences followed by part of speech tagging. Thereafter, the remaining modules identify the time expression and send to a discourse processing module that is capable of solving context dependent time expressions. GUTime is another temporal tagger which adds TIMEX3 tags and has been evaluated on TERN 2004 training corpus to get an F-measure scores of 0.85 and 0.82 for temporal expression recognition and normalization. Another rule based temporal extractor is Heideltime and SuTime. While SuTime is optimized for English texts, Heidel time is designed to support multi languages that is adapted to both news domain and narrative documents. Both these have achieved similar scores in extraction process with Heideltime lagging by an F Score of 0.002. But, Heideltime had the better normalization process with an F measure of 0.776 compared to SuTime's 0.674. In this research analysis of temporal information using temporal taggers like TarSqi toolkit and Heideltime that can be described below.

Timeml and Tarsqi Toolkit

TimeML is the time markup language. It is based on XML-based markup language for encoding temporal and event-to-time relations for use in automatic text processing. However, it can be used more extensively in annotating news articles, TimeML has also been used to extract time expressions from e-mails, legal documents, and medical texts [19]. The primary goals of this are the ordering events, time stamping events, reasoning about the duration of events and contextual reasoning of "underspecified time expressions" [20]. The building blocks of the TIMEML are events, times, and the temporal relationship between time and events. It uses four different types of XML tag types: TIMEX3; EVENT; SIGNAL: and LINK:

The TIMEX3 tag is based on the TIMEX2 time annotation schema. It focuses on the identification and normalization of the temporal expressions. TIMEX2 annotation schema identifies the temporal expressions and then normalizes to the standard format like ISO 8601 numerical representation of dates and times when possible. Several properties can be included in the temporal expressions namely granularity, indexicality, context dependence, ambiguity, and fuzziness of boundaries that are taken into account in the annotation scheme.

Example of TIMEX2 tags are:

1) <TIMEX2 VAL = "2013-09-08T10:30:16"> September 08, 2013 10:30:16 EST </TIMEX2>
2) <TIMEX2 VAL= "2013-FA"> Fall 2013 </TIMEX2>
3) <TIMEX2 VAL="2013-08-15NI"> last night </TIMEX2>, refers to August 15, 2013
4) <TIMEX2 VAL="2013-08-19">five days ago </TIMEX2>, refers to August 15, 2013

5) <TIMEX2 VAL=”2013-W29”> Next week </TIMEX2>, refers to August 15, 2013

TIMEX2 tags used a rule-based approach, whereas a series of perl regular expressions available for identification of specific patterns of temporal expressions. This use of regular expression matching and substitution places TIMEX2 tags around any occurrence of an apostrophe followed by double digits. The tag is TIMEX2 of type date,

e.g. <TIMEX2 TYPE=”DATE”> ’99 </TIMEX2>. TIMEX3 developed based on the work TIMEX2.

According to TimeML guidelines, TIMEX3 captures four types of temporal expressions: DATE (for calendar date); TIME (for time of day); DURATION; and SET (for reoccurring times). [21]

Examples of TIMEX3 tags are:

1) <TIMEX3 tid=”t1” type=”DATE” value=”2014-11-2”> November 2, 2014 </TIMEX3>
2) <TIMEX3 tid=”t2” type=”DURATION” value=”P3D”> four days </TIMEX3>
3) <TIMEX3 tid=”t3” type=”SET” value=”P1W” quant=”EACH” freq=”3D”> 4 days each week </TIMEX3>

Where tid is the time ID number of the expression, and value holds the normalized. ISO 8601 format of the temporal expression if calculable. X are usually placeholders and are used for expressions which can only be partially specified, e.g. “XXXX-02-03” for February 3 and no year. It must be noted that it was TIMEX3 introduced the notion of a “temporal anchor” so that we can accommodate expressions whose temporal reference cannot be retrieved by evaluating the expressions themselves. The anchor then provides the temporal reference for such underspecified expressions via an anchor time ID. Other expressions, such as event or date/time expressions, may serve as anchors.

i) EVENT tags

EVENT tags are divided into seven classes: reporting, perception, aspectual, I_action, I_state, state, and occurrence. [21] These classes primarily represent different verbs seen within sentences. Multiple instances of events are tagged by MAKEINSTANCE tags.

ii) SIGNAL tags

SIGNAL tags identify temporal prepositions and conjunctions such as *before, after, during, since, until, at, on, in, for, over, throughout, while*, and *when*. Such terms act to place temporal expressions and the events associated with them on a timeline. An example of SIGNAL tagging for “on October 15, 2004” is : <SIGNAL sid=”s1”> on </SIGNAL> <TIMEX3 tid=”t1” type=”DATE” value=”2004-10-15”> October 15, 2004 </TIMEX3>

iii) LINK tags

TimeML has 3 LINK tags: TLINK; SLINK; AND ALINK, all of which provide linking or relational information. SLINK and ALINK tags convey subordinating and

aspectual relationships, respectively, between events. "TLINK [Temporal LINK] tags are arguably the most important tag in all of TimeML". [26] The purpose of the tag is to provide a classifier which relates two intervals of time to one another. These relations determine the anchoring and ordering of events within a narrative's timeline. TLINK's rule set takes into account several syntactic rules based on intra-sentential presence of SIGNAL tags, event types, verbs, and verb tenses. The relation types between time intervals and events are similar to Allen's 13 interval relationships: *simultaneous;*

before; after; immediately before; immediately after; including; being included; during; beginning; begun by; ending; ended by; and *identity.* The TTK uses constraint propagation to ensure the consistency of all relations.

TARSQI Toolkit (TTK)

Temporal Awareness and Reasoning Systems for Question Interpretation was developed for enhancing natural language question answering systems. In the news articles contains temporally based questions about the entities and events. In some instances, a Keyword based approach is inadequate to answer some temporal questions like "Did India land on the Moon in 2007?" In order to automatically extract events and annotate temporal information a modular system was developed. It is used to identify the events and temporal expressions in the natural language texts, also parse the document to order events and to anchor them to temporal expressions. The TTK architecture consists of several components are Treetagger for preprocessing the documents, GUTime for extracting the temporal expressions, and Evita for identifying the events. The additional components are used to link the time and events, and also propagate the links use the Allen's algorithm for transitive closure like Slinket, Blinker, S2T, Tlink Classifier, SputLink and Link merger. Time tagger based on the TIMEX3. For temporal ordering, TTK a complete toolkit available for free use [22]. The Time Markup-Language, (TimeML), specifies the types of temporal expressions and events to annotate within documents and what XML tags to use. Based on TimeML specification, the TTK software package was created to perform the following steps: Identify temporal expressions, Place numerical values to temporal expressions, Identify events, Link times and events together. The numerical values assigned to the temporal expressions based on the accuracy of the first step that is identification of the temporal expressions. Identification of temporal expressions includes the two steps that are recognizing patters of temporal expressions through the use of a series of perl regular expressions, and identifying the type of the temporal expressions found in the documents. TimeML specifications it can be classified into four types, DATE (for calendar date), TIME (for time of day), DURATION (for spans of time), SET (for recurring times).

Heideltime:

Heideltime is a cross-domain, multilingual temporal tagger that extracts temporal expressions from the documents and normalizes the temporal expressions based on the TIMEX3 annotation standard. It is mainly used to normalize the documents in different domains like news, colloquial (e.g., SMS, tweets), narratives (e.g.,

Wikipedia articles), and scientific (e.g., biomedical studies). Heideltime is a rule based system, and the resources like normalization information, pattern, and rules are strictly separated from the source code due to its architectural feature. Heidel time supports eleven languages like English, Spanish, German, Dutch, Italian, Arabic, Vietnamese, Chinese, Croatian and Russian. Heideltime uses two different language resources that are language-dependent resources like patterns, rules, and normalization information and language independent but that are likely to be domain-sensitive normalization strategies. Heideltime, separates the language dependent resources from the algorithmic part. Language dependent resources consists of three parts. The first pattern resources it includes the frequently used patterns to form the temporal expressions. Example for this, separate pattern file that contain name of months for each language. Second, normalization resources that contains mapping from patterns to their normalized values for example "April" is normalized to "04". Finally the rule resources, it contains several rules that are used to define the how pattern resources are combined for the extraction of temporal expressions and the extracted patterns are normalized to standard format, how this extracted patterns combined to their normalization resources. Heideltime also applied to the domain-dependent normalization strategies [23]. In the domain-dependent areas that can be addressed different handling of 2-digit year expressions in news and narrative-style documents.

The temporal expressions contains a three-tuple TE= <e, t, and v> where e is the temporal expressions that occur in the document, t is the type of the temporal expressions, v is the value of the temporal expressions that is normalized value. There are four possible types Date, Duration, Time, and Set. The main goal of the Heideltime is to identify the temporal expressions and to set the type of the temporal expressions and finally to assign the value attributes.

Evaluation of The Dataset:

Samples:

TimeML developers also collected and annotated the Timebank v1.2 Corpus. This corpus consisted of 18359 news articles from a variety of sources, including Public Radio International (PRI), Voice of America (VOA), ABC, CNN, Associated Press (AP), New York Times (NYT), and the Wall Street Journal (WSJ). Each article was "annotated following the TimeML 1.2.1 specifications" [24]. According to Timebankv1.2 documentation, annotators had "various linguistics background intimately familiar with the latest specifications" [36]. The Corpus was available free of charge from the Linguistic Data Consortium (catalog ID LDC2006T08 and ISBN 1-58563-386-0). [25] This collection was non-annotated, but was in appropriate XML format, ready for input and processing by TTK and HeidelTime.

Measurements:

Temporal expressions that identified from the documents news articles are as follows:

DATE (for calendar date):

1. Includes specialized time patterns such as “12/2/2009”, “11-01”, and “03/2014”.
2. Includes proper nouns such as days of the week (“Monday”, “Tuesday”, “Wed”); months of the year (“January”, “Feb”); holidays (“Thanksgiving Day”, “May Day”), and seasons.

TIME (for time of day):

1. Includes specialized time patterns such as “8 PM”, “30/6/2014 0400”, and “12 o’clock”.
2. Includes nouns such as “morning”, “evening”, “afternoon”.

DURATION denotes a span of time at varying levels of granularity:

1. Includes expressions containing nouns such as “seconds”, “minutes”, “days”, “weeks”, and “years”, denoting a span of time, e.g. “for three weeks”, “over a number of days”, “throughout the year”.
2. Does not include expressions such as “in two days”, “in twenty hours”, “3 weeks ago”, which were categorized as date or time instead.

SET (for recurring times i.e. periodicity):

1. Includes expressions containing adverbs and adjectives such as “daily”, “every day”, “per week”, and abbreviations such “qd”, “b.i.d”, and “t.i.d”.

Methodology

Non-annotated documents were processed by TTK and Heideltime for the automated recognition of temporal expressions. The Tarsqi toolkitv1.0 obtained from[22]. TTK’s developers developed and tested it for Red Hat Linux 5, with Python 2.4.3 and Perl 5.8.8; and for the Mac OS X, with Python 2.3.5 and Perl 5.8.8. First, installed VMware Workstation 6.5 on a Windows 8 host operating system to run a virtual machine with Red Hat Enterprise Linux (RHEL) 5 Desktop as a guest operating system. The virtual network adapter to use Network address translation (NAT) was configured in order to share the internet protocol (ip) address and files of the host computer. Then TTK was installed onto the RHEL 5 guest operating system following instructions given by the TimeML website manual [22].

Table 1: Time expressions found in a document using TTK

TIMEX	TYPE	VALUE
Last twenty four hours	DURATION	PT24H
Five year	DURATION	P5Y
Four year	DURATION	P4Y
Once	DATE	PAST-REF
Long ago	DATE	PAST-REF
now	DATE	PRESENT-REF
October	DATE	2001-10
now	DATE	PRESENT-REF
The past three months	DURATION	P3M
Ten years	DURATION	P10Y

Similarly Heideltime can be downloaded from the [26] and then installed into the Red Hat Enterprise Linux (RHEL) 5.

A set of 15000 documents were processed first in TTK toolkit and then in Heidel time. Table 2 provides the expressions retrieved from a document using Heideltime, whereas Table 3 compares it with TARSQI ToolKit.

Table 2: Time Expressions found in a document using Heidel Time

TIMEX	TYPE	VALUE
The last twenty four hours	DURATION	PID
Five year	DURATION	P5Y
Four year	DURATION	P4Y
now	DATE	PRESENT-REF
October	DATE	2001-10
Now	DATE	PRESENT-REF
The past three months	DURATION	P3M
Ten years	DURATION	P10Y
A couple of years	DURATION	PXY

Table 3: Comparison of Temporal annotations of Heideltime and TTK

Heideltime annotations	**Types**	**TTK annotations**	**Types**
The last twenty four hours	DURATION	The last twenty four hours	DURATION
Five year	DURATION	Five year	DURATION
Four year	DURATION	Four year	DURATION
now	DATE	once	DATE
October	DATE	Long ago	DATE
Now	DATE	now	DATE
The past three months	DURATION	October	DATE
Ten years	DURATION	Now	DATE
A couple of years	DURATION	The past three months	DURATION
		Ten years	DURATION

15000 documents form the TimeML dataset were processed at the rate of one thousand documents at a time and a table of extracted information is shown below for a given Heideltime annotated expression. A corresponding TTK annotated expression is matched in the following manner.

Table 4: Annotated Expressions

Category	**String Match**	**Type (date, time, duration, set) match**
1	Exact Match	Match
2	Exact Match	Not Matched
3	Overlap (Partial Match)	Match
4	Overlap (partial Match)	Not Matched
Miss	No match	

The following annotations are used below.

1. Correct – The compared items are identical
2. Incorrect – The compared items are not identical
3. Miss – A reference had No TTK output
4. Spurious - TTK output had no reference in Heidel time
5. Possible – Correct + Incorrect + Miss
6. Actual – Correct + Incorrect + Spurious
7. Recall – Correct/Possible
8. Precision – Correct/ Actual
9. F-Measure = $\frac{2 * Precision * Recall}{Precision + Recall}$

The below tables shows all the values of TTK compared against Heidel time. The evaluation parameter values obtained are shown in Fig 4.

Table 5: TTK Scores against Heidel Time

	Date	Time	Duration	Set	Total
CORRECT	9768	486	1062	663	11979
INCORRECT	486	177	487	442	1592
MISS	1635	398	708	265	3006
SPURIOUS	928	265	89	133	1415
POSSIBLE	11890	1061	2254	1370	16575
ACTUAL	11183	928	1639	1238	14998
RECALL	0.82	0.45	0.47	0.48	0.72
PRECISION	0.87	0.52	0.64	0.53	0.79
F-MEASURE	0.84	0.48	0.54	0.50	0.76

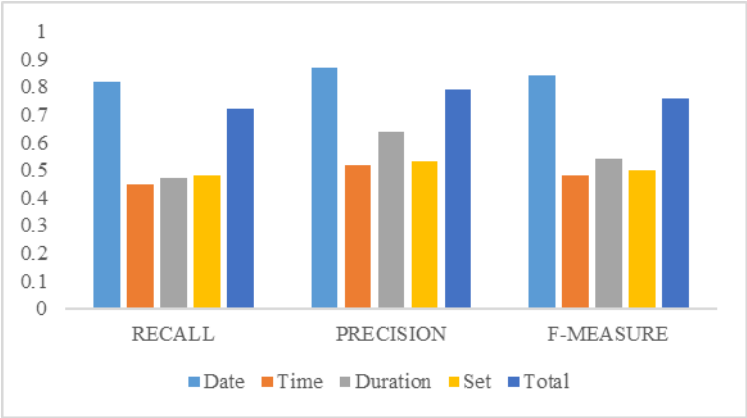

Figure 4: Evaluation Parameter Values of Extracted Temporal Expressions

Conclusion and Future Works

Based on the above analysis of Heidel Time and TTK, it was found that Heidel Time was partially more effective in retrieving temporal expression than TTK toolkit. The precision value was measured for a set of 15000 documents. The precision values of date, time, duration and set came out to be 0.87, 0.52, 0.64 and 0.53 respectively leading to an overall precision value of 0.79 and the recall value for the same was found to be 0.82, 0.45, 0.47 and 0.48 respectively with an overall recall value of 0.72. When compared with Heidel time, the F-Measure for the same was 0.84, 0.48, 0.54 and 0.50 respectively with an overall value of 0.76. It was also found that generally, the temporal expression type “Date” was dominantly present in the dataset compared to other types such as Duration, Time etc. The same trend was picked up by both HeidelTime and Tarsqi.

There is much work needed to improve the TTK performance such as recognition and retrieval of Sets as it is a region where it clearly lags behind HeidelTime. Evaluations for the accuracy of the values placed on other types of temporal

expressions other than Sets like Date, time and Duration are needed. Also identification of temporal expressions by using anchor points need to be explored.

Acknowledgement

The authors of this paper would like to acknowledge Marc Verhagen for his timely help regarding TARSQI.

References

[1] Marc Verhagen and James Pustejovsky 2008 Temporal Processing with the TARSQI Toolkit, Coling 2008:comanion volume-Posters and Demonstrations,pages 189-192

[2] Jannik Strotgen and Michael Gertz HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 321–324.

[3] Bruce, B. C. 1972. A Model for Temporal References and its Application in a Question Answering Program. Artificial Intelligence, 3: Elsevier. 1-25.

[4] Allen, F. J. 1983. Maintaining Knowledge about Temporal Intervals. CACM: Communications of the ACM, 26(11): ACM Press. 832-843.

[5] Snodgrass, R. and Ahn, I. 1985. A Taxonomy of Time Databases. In Proceedings of the SIGMOD'85. Austin, Texas, USA. May 28 - 31: ACM Press. 236-246.

[6] Setzer, A. and Gaizauskas, R. J. 2000. Annotating Events and Temporal Information in Newswire Texts. In Proceedings of the LREC'00. Athens, Greece. May 31-June 2: ELDA.

[7] Allan, J., Carbonell, J., Doddington, G., and Yamron, J. 1998. Topic Detection and Tracking Pilot Study Final Report. In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. Lansdowne, Virginia, USA. February. 194-218.

[8] Swan, R. and Allan, J. 1999. Extracting Significant Time-Varying Features from Text. In Proceedings of the CIKM'99. Kansas City, USA. November 2-6: ACM Press. 38-45.

[9] Makkonen, J. and Ahonen-myka, H. 2003. Utilizing Temporal Information in Topic Detection and Tracking. In ECDL'03. Lecture Notes in Computer Science, 2769/2004: Springer-Verlag. 393-404.

[10] Shaparenko, B., Caruana, R., Gehrke, J., and Joachims, T. 2005. Identifying Temporal Patterns and Key Players in Document Collections.

In Proceedings of the TDM'05 Workshop associated to ICDM'05. Houston, USA. November 27-30: IEEE Computer Society Press. 165-174.

[11] Setzer, A. and Gaizauskas, R. J. 2000. Annotating Events and Temporal Information in Newswire Texts. In Proceedings of the LREC'00. Athens, Greece. May 31-June 2: ELDA.

[12] Jatowt, A. and Yeung, C. M. 2011. Extracting Collective Expectations about the Future from Large Text Collections. In Proceedings of the CIKM'11. Glasgow, Scotland, UK. October 24-28: ACM Press. 1259-1264.

[13] Nunes, S., Ribeiro, C., and David, G. 2008. Use of Temporal Expressions in Web Search. In Proceedings of the ECIR'08. Lecture Notes in Computer Science, 4956/2008: Springer-Verlag.580-584.

[14] Jones, R. and Diaz, F. 2007. Temporal Profiles of Queries. ACM Transactions on Information Systems, 25(3). Article No.: 14.

[15] Costa, F. 2013. Processing Temporal Information in Unstructured Documents. PhD thesis, Universidade de Lisboa. May 31. 1-281.

[16] Mani, I. and Wilson, G. 2000. Robust Temporal Processing of News. In ACL'00. Hong Kong, China. October 1-8: Association for Computational Linguistics. 69-76.

[17] Strötgen, J. and Gertz, M. 2010a. HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions. In Proceedings of the IWSE'10 associated to ACL'10.Uppsala, Sweden. July 11-16. 321-324.

[18] Chang, A. and Manning, C. 2012. SUTIME: A Library for Recognizing and Normalizing Time Expressions. In Proceedings of the LREC'12. Istanbul, Turkey. May 23-25

[19] Schloss Dagstuhl Seminar Homepage. Annotating, Extracting and Reasoning about Time and Events. http://www.dagstuhl.de/en/program/ calendar/semhp/?semnr=05151

[20] TimeML.Markup Language for Temporal and Event Expressions. http://www.timeml.org/site/index.html

[21] Pustejovsky J, Knippen R, Littman J, Sauri R. Temporal and Event Information In Natural Language Text. 2005.

[22] http://www.timeml.org/site/tarsqi/toolkit/download.html

[23] Strötgen, J. and Gertz, M. Proximity2-aware ranking for Textual, Temporal, and Geographic Queries. In Proceedings of CIKM'13. San Francisco, USA. October 27-November 01: ACM Press, 739-744.

[24] TimeML Corpora http://www.timeml.org/site/timebank/ timebank.html

[25] TimeBank 1.2 http://www.ldc.upenn.edu/Catalog/ CatalogEntry.jsp? catalogId=LDC2006T08

[26] https://code.google.com/p/heideltime/