Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2015) 23: 212 – 230

doi:10.3906/elk-1208-93

Temporal logic extension for self-referring, nonexistence, multiple recurrence, and anterior past events

Şadi Evren ŞEKER*
Department of Computer Engineering, İstanbul University, Avcılar Campus, Avcılar, İstanbul, Turkey

Received: 29.08.2012 • **Accepted:** 03.04.2013 • **Published Online:** 12.01.2015 • **Printed:** 09.02.2015

Abstract: This study focuses on the possible extensions of current temporal logics. In this study, 4 extensions are proposed: self-referring events, nonexisting events, multiple recurrence of events, and an improvement on anterior past events. Each of these extensions is on a different level of temporal logics. The main motivation behind the extensions is the temporal analysis of Turkish. Similar to temporal logic studies built on other natural languages, like French, Ukrainian, Italian, Korean, English, or Romanian, this is the first time that the Turkish language has been deeply questioned in the sense of computable temporal logic using the view of a standardized temporal markup language. This study keeps the methodology of TimeML and researches Turkish from the perspectives of Reichenbach and Allen's temporal logics. Reichenbach's temporal logic is perfectly capable of handling the anterior temporal feeling, but it is not enough to handle the sense of 'learnt' or 'study', which are 2 past tenses in Turkish. Moreover, Allen's temporal logic cannot handle 2 events following each other continuously, which is called recurring events in this study for the first time. Finally, based on the experiences from a 4-year PhD study on natural language texts, this study underlines the absence of self-referring or a reference to nonexisting events in temporal logics. After adding the above extensions to computable temporal logic, the capability of tagging the events in Turkish texts is measured with an increase from 18% to 100%, creating a Turkish corpus for the first time. Moreover, new software is implemented to visualize the tagged events and previous software is developed to handle events tagged for Turkish.

Key words: Natural language processing, temporal logics, Allen's temporal logic, TimeML, Reichenbach's temporal logic

1. Introduction

Natural language processing is a recently popular research area in computer science [1–3], which can be integrated with temporal logics that have a wide range of use in many studies, such as real-time systems, multimedia applications, security, and so on [4–8]. Moreover, studies on temporal logics have a theoretical background like the satisfiability degree [9], symbolic transition systems [10], synthesizing [11], fuzzy temporal logics [12], or aspect-oriented verifications [13].

The aim of this study is to achieve a machine-computable temporal logic that covers Turkish temporal cases. To the best of the author's knowledge, there is no specific research on the formal representation of temporal semantics in Turkish.

On the other hand, for some natural languages, some studies already declared temporal differences as affected by their culture and, therefore, their languages. For example, TETI for the Italian language [14], ISO-

*Correspondence: academic@sadievrenseker.com

Time Markup Language (TimeML) for French TimeBank [15], TimeML for the semantic web [16], TimeML and Romanian case [17], KTimeML for Korean case [18], Spanish text analysis [19], Chinese temporal modelling [20], or extensions on linear temporal logics by adding "since" and "previous" [21] are only a few of these machine-computable temporal logic applications.

In this study, the parts common to and different among Turkish temporal logic and other natural language-oriented temporal logics are defined. After the results of these differences are found, TimeML [22–24] is extended to cover the Turkish language. Moreover, a corpus is collected in Turkish and the success of the extended version of TimeML is tested over this corpus.

This paper is organized as follows. Section 1 covers the basic concepts followed by predefinitions in Section 2, and Section 3 covers the concept of time tagging. The 2 major temporal logics, Reichenbach's temporal logic (RTL) [25–29] and Allen's temporal logic (ATL), are explained before TimeML. Moreover, in the literature, both ATL and Allen's interval logic (AIL) are well accepted, and in this paper, ATL will be used. After the TimeML, Turkish temporal logic is analyzed and compared with ATL and RTL, and some extensions of the TimeML are suggested to cover these differences.

Finally, the success of these extensions is tested over a corpus collected randomly from children's stories specifically for this study. Unfortunately, the corpus is newly gathered for this study, since no previous study exists in the field. Readers will find a comparison between the original TimeML and the extended version of TimeML for Turkish.

2. Predefinitions

This section covers the definition of 'time' from the perspective of this study. The word 'time' may have many definitions, depending on the scope. From the scope of temporal logic, the word 'time' can be defined as a reference to the events. From our perspective, the time of an event can only be interpreted by reference to another event.

In the literature, almost all temporal logics separate temporal statements into 2 distinct groups:

1. Absolute time, e.g., "I played tennis on October 19, 2010, at 3:38 pm".
2. Relative time, e.g., "I played tennis after I left school and arrived at the tennis court".

The difference between the 2 above statements is the event to which they refer. The first temporal statement refers to events in the solar system, like the position of the Earth in relation to the sun, or historical events, like the birth of Christ. Therefore, in this study, I only define the time of an event as a reference to another event. Aside from the above definition, I also accept that all the events occur in temporal systems. In other words, no event can exist outside of the time realm. Some elements of the time of the event may be unknown, but all of the events are still connected to time, and therefore, connected to another event.

The last temporal statement is the sense of time. This study does not cover the aspect of humans' sense of time. Time sense can be defined as the feeling of time during any event, which can differ according to the individual.

For example, the sense of how a minute feels can change according to whether a subject is putting their hand on a flame or eating their favorite dessert. In both cases, the time spent can be measured in minutes but the feeling of time differs for the same person, or the sense of duration may differ for each person for the same event. For example, one person may feel as if an exam takes a minute, while another may define the duration of the same exam as centuries. These examples are included here simply to illustrate that 'sense of time' is intensely subjective; it cannot be formally processed by computers using the current technology, and, thus, I exclude sense of time from this study.

3. Time tagging

Time tagging, which can be interpreted as the extraction of temporal semantics from natural language texts, is mainly used in almost all natural language research areas, like question answering, text summarization, or visualization of texts [30,31].

A natural language document contains semantic values from many different categories. For example, locations [32], personal information [33] or know-how [34] can all be passed through natural language sentences. All of this semantic information in natural language sentences can be expressed using a temporal logic, since all verbs are strongly connected to time [35]. For example, a sentence without location information is valid, but all sentences are considered existing within a time boundary.

Time tagging is a technique for marking temporal information of events, time expressions, or the relations of events on a timeline. Although the linearity of time is an open discussion, time tagging techniques are only built over linear temporal logics. For example, AIL [36] or RTL [37] are 2 representations of time in a linear form.

After the correct tagging of a natural language text, the implementation of some automated codes can process the event by ordering or answering questions. The tagging can be done in 2 possible ways. For mature natural language processing (NLP) languages, which mostly solve morphological and syntactic parsing problems, it is possible to implement autonomous software to tag temporal information about the NLP. On the other hand, for languages still under massive development of NLP and that do not offer satisfactory results on morphological and syntactic levels, the temporal information can be tagged only manually.

Time tagging is still important for these languages in order to target a purpose for the representation of semantic information after syntactic studies, and to prepare tools for use after achieving desired NLP levels. For example, in Turkish, no satisfactory syntactic and morphological tool exists to extract semantic information. Furthermore, temporal logic in Turkish is different than in Indo-European languages in some ways.

After researching temporal logics built in reference to the Indo-European language family, I have figured out the temporal differences in Turkish and also developed a representation tool. This article presents the results.

3.1. Reichenbach temporal logic

RTL is built on 3 simple temporal anchors:

- Speech time (symbolized by S),
- Reference time (symbolized by R),
- Event time (symbolized by E).

Most of Reichenbach's study was focused on natural languages. Thus, he formulated the order of these 'times'.

For example, a sentence like "She read the book" can be formalized as $R = E < S$. On the other hand, a sentence like "I have read the book" can be formalized as $E < R = S$. Please note that in the former model, the event takes place before the speech time and the speech refers to the event time, so the event and reference times are equal and smaller than speech time in the model. For the latter example, the event again takes place before the speech time, but the speech is referring to the current time, so the speech time and reference times are equal and greater than the event time.

By a simple probability calculation, 13 possible orders for the above temporal anchors can be listed. Obviously, not all of these probabilities are meaningful in a natural language. Reichenbach named these possibilities using anterior, simple, and posterior aspects, and past, present, and future tenses. According to Reichenbach, there can be only 9 possible meaningful times in English or in any natural language. Table 1 covers these possibilities and gives samples for each of the cases.

Table 1. All 13 of the possible permutations of RTL, their English tenses/aspects, and a sample of each case.

Permutation	Reichenbach tense name	English tense	Sample
E < R < S	Anterior past	Past perfect	I had slept
E = R < S	Simple past	Simple past	I slept
R < E < S			
R < S = E	Posterior past		I would sleep
R < S < E			
E < S = R	Anterior present	Present perfect	I have slept
S = R = E	Simple present	Simple present	I sleep
S = R < E	Posterior present	Simple future	I will sleep
S < E < R			
S = E < R	Anterior future	Future perfect	I will have slept
E < S < R			
S < R = E	Simple future	Simple future	I will sleep
S < R < E	Posterior future		I shall be going to sleep

Please note that in Table 1, blank lines represent meaningless cases of these permutations in English.

3.2. Allen's interval logic

ATL deals with orders of events. The representation of event orders like "event A is before event B" or "event A is at the same time as event B" are the operators of this logic.

The basic variables in AIL are the intervals and Allen has built his logic over binary operators working on those intervals. In AIL, 13 basic binary operators connect intervals by constraints. These intervals can be considered to be running threads or any operations on the timeline.

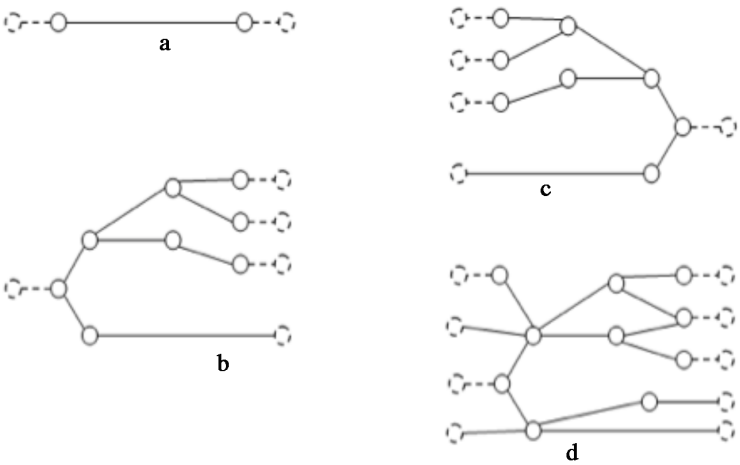

Figure 1. Linearity of the timeline.

The linearity of time can vary between different temporal logics. Figure 1 displays 4 different types of timeline linearity. Figure 1a is linear in the past and future. Figure 1b is only linear in the past and is nonlinear in the future. This type of linearity can be classified as semilinear. Figure 1c is the opposite form of Figure 1b, where the future is linear and the past is nonlinear. Figure 1d is nonlinear in both the past and the future.

ATL supports all types of linearity in Figure 1. ATL can be demonstrated by a box diagram, where the boxes represent events and the arrows represent relations between them.

For example, in a sentence like "John ate an apple at the table after he entered the room", there are 2 events: "eat" and "enter". There are also hidden events, in which John goes to the table and takes the apple, in order to eat an apple from the table after he has entered the room. The timeline of the example is given in Figure 2.

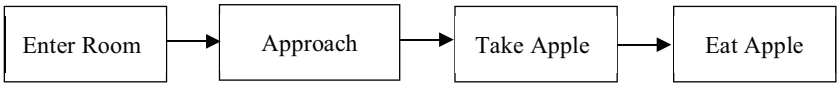

Figure 2. Sequence of events in the sample sentence.

If the states of the events are considered, it is known that John was outside the room before he entered the room. He was also away from the table before he approached the table, and he had no apples before taking the apple.

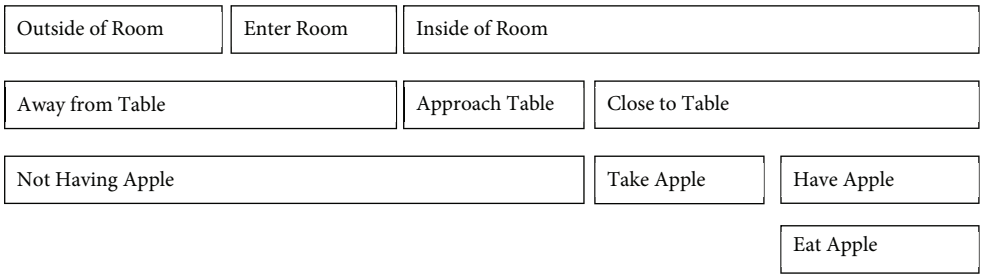

Figure 3. Events and states diagram.

Therefore, from the sample sentence demonstrated in Figure 3, it is possible to conclude that John had an apple when he was inside the room or that John had no apple while he was either away from the table or while he was outside the room.

The 13 types of possible relations and the possible operators of AIL are listed below.

- Before (x,y) or After (y,x)
- Overlaps (x,y) or Overlapped (y,x)
- Meets (x,y) or Met by (y,x)
- Contains (x,y) or During (x,y)
- Starts (x,y) or Started by(y,x)
- Ends (x,y) or Ended by (y,x)
- Equals (x,y)

The above sample can be modeled using ATL. Let us say that entering the room (ER) requires us to be outside of the room (OR); after entering the room, the state is inside the room (IR) and, similarly, approaching the table (AT) changes the state of being away from the table (SAT) to the state of being close to the table (SCT). Taking an apple (TA) is a transformation of state from not having the apple (NHA) to having the apple (HA). All these states are prerequirements in the case of eating the apple (EA).

The above sentence can be modeled in ATL in the following formulation:

$$\begin{array}{lllllll} \textit{Meets(OR,ER)} & \wedge & \textit{Meets(ER,IR)} & \wedge & \textit{During (ER,SAT)} & \wedge \\ \textit{During (AT,IR)} & \wedge & \textit{Meets(SAT,AT)} & \wedge & \textit{Meets(AT,SCT)} & \wedge \\ \textit{During(AT,NHA)} & \wedge & \textit{During(TA,IR)} & \wedge & \textit{During(TA,SCT)} & \wedge \\ \textit{Meets(NHA,TA)} & \wedge & \textit{Meets(TA,HA)} & \wedge & \textit{During(EA,HA)} & \wedge \\ \textit{During (EA,CT)} & \wedge & \textit{During(EA,IR)} & \wedge & \textit{Meets(TA,EA)} & \end{array}$$

The formulation above demonstrates all of the temporal states and events in the sample sentence. On the other hand, a reader can interpret the above sentence and can add more states, which can still be modeled by ATL. For example, if John follows the order of events above when he is hungry, then this state can be added to the model of ATL. In this case, the model would be:

Occurs (hungry, NHA) $\wedge$ Holds (hungry, TA) $\wedge$ Meets (hungry, EA)

Subsequently, from the ATL model, John eats an apple when he gets hungry and does not have an apple; he takes an apple while he is hungry; his state of hunger ends when eating the apple.

3.3. TimeML as a time markup language

The TimeML is a time markup language, first developed in 2002, which has recently become a very popular standard for stamping events, ordering events, reasoning about persistence of events, and modeling time expressions in natural language documents.

Obviously, the TimeML was developed for English [38], and now some other languages like Ukrainian [39] or French [40] have also been applied to the TimeML, with some modifications.

The latest version of the TimeML, modified in 2006 [22–24], has 3 layers of semantic representation:

- Event level,
- Signal level,
- Link level.

In the event level, the events in natural language documents are stamped with information about the event, like the tense, aspect, modal, continuity, or plurality. In the signal level, those events are marked with the recurrence or durations. Finally, on the top-most level, the link level, those signaled events are connected with before, after, eventually, and similar connections [40].

The connection type of the TimeML in the link layer was inspired by Reichenbach's temporal modeling [41], which, unfortunately, does not support a full Turkish temporal logic, as explained in the following section. The tenses of events in the TimeML are defined in < MAKEINSTANCE > tags, which are mainly designed for creating real-time events from theoretically defined < EVENT > tags. The relationship between MAKEINSTANCE and EVENT tags is similar to the relationship between object and class in object-oriented programming, where EVENT tags can be considered as classes and MAKEINSTANCE as objects.

The tense and aspect value of the event are parameters of the MAKEINSTANCE tags, which are listed below:

```
tense::  = 'PAST'|'PRESENT'|
'FUTURE'|'NONE'
|'INFINITIVE'|'PRESPART'
|'PASTPART'
aspect::  = 'PROGRESSIVE'|'PERFECTIVE'
|'PERFECTIVE_PROGRESSIVE'
|'NONE'
```

The tense possibilities for PAST, PRESENT, and FUTURE exactly match RTL. On the other hand, NONE defines an event without a tense. Moreover, in the TimeML, verbs are categorized by meanings. For example, a verb that has a current ongoing effect and a verb that finished its effect are separated. This separation requires adding the above infinitive, pre-part, and past-part alternatives. For example, "release" is infinitive, "seeking" is present participle (coded as 'pre-part'), and "found" is past participle ("past-part"). The most crucial point here is that the design of all above tenses and aspect possibilities was carried out according to English temporal logic.

4. Turkish temporal logic

Turkish is a member of the Ural-Altaic language family [38]. This section analyzes Turkish from a language perspective and illustrates its important aspects. Turkish is characterized by certain morphophonemic, morphtactic, and syntactic features: vowel harmony, agglutination of all-suffixing morphemes, free order of constituents, and head-final structure of phrases.

The Turkish language uses Latin characters, and the success of NLP studies on its morphological and syntactic levels is still very low, especially when compared with the success rates of studies on Indo-European languages such as English. Because of this fact, most researchers working on NLP are quite far from concentrating on formal semantic level studies. Part of this study focuses on Turkish temporal logic, which is concentrated on the semantic representation of Turkish texts and points out a possible path for researchers by achieving success in representing the morphological and syntactic levels of Turkish.

4.1. Reichenbach modeling and the TimeML of Turkish temporal logic

Table 2 covers the application of RTL to English. It includes empty lines for meaningless cases in English.

Despite having no future tense but modals that translate into English, Turkish has a variety of future tenses [41]. Please note that the translation column is prepared only to give information about meanings for non-Turkish readers. The translations do not hold the exact temporal feeling as in Turkish. In Turkish, some cases in Table 3 are used rarely in daily life. For example, the cases with 2 words, in which the second word starts with 'ol-', are used very rarely in Turkish temporal logic, but are still meaningful and understandable for any Turkish speaker.

The grammatical term 'hikaye' can also be translated into English as 'storytelling'. This alternative will be named 'story' and this temporal form can be represented as anterior past in RTL, where the only semantic addition is the subject of the event does the event. The word 'rivayet' can be translated into English as 'reporting', where the subject does not do the event but rather learns about it from somebody else. This alternative is named 'learned' to avoid a mixture of reporting aspects in the links of the TimeML.

In Turkish semantics, the storytelling tense requires someone to live the event and reporting requires someone to learn, so these tenses are named 'story' and 'learned'. The auxiliary verbs in the above examples

starting with 'ol-' translate into 'being' in Turkish. Therefore, the third tense alternative suggested for TimeML is a 'BEING' alternative.

Table 2. All 13 of the possible permutations of RTL and their Turkish tense/aspect and a sample for each case.

Permutation	Reichenbach tense name	Turkish tense	Sample	Translation to English
E < R < S	Anterior past	Geçmişin hikayesi	Uyumuştum	I have slept
E = R < S	Simple past	Geçmiş	Uyudum	I slept
R < E < S		Gelecek hikayesi	Uyuyacaktı	He was planning to sleep
R < S = E	Posterior past			
R < S < E		Gelecek rivayeti	Uyuyacakmış	I learnt that he will sleep
E < S = R	Anterior present	Şimdiki rivayet	Uyumuşum	I have been sleeping
S = R = E	Simple present	Şimdiki	Uyuyorum	I am sleeping
S = R < E	Posterior present	Gelecek	Uyuyacağım	I am going to sleep
S < E < R				
S = E < R	Anterior future		Uyuyor olacağım	I will be sleeping
E < S < R			Uyumuş olacağım	I will have been sleeping
S < R = E	Simple future	Gelecek zaman	Uyuyacağım	I will sleep
S < R < E	Posterior future		Uyuyacak olacağım	I am going to be sleeping

Table 2 can be modified to show Turkish temporal logic, as shown in Table 3.

Table 3. Application of 'story' and 'learned' differences to RTL.

Relation	Sample	Translation to English	Relation	Sample	Translation to English
ST,E < R < S	Dün geleceğim dediydi	He said that he has already come yesterday	LR,S = R = E	Geliyorum diyor	He is telling that he is coming
LR,E < R < S	Dün geleceğim demişti	He had said that he had already come yesterday	ST,S = R < E	Şimdiye dönecek	He will return now
ST,E = R < S	Geliyorum dedi	He said he is coming	LR,S = R < E	Şimdiye dönecek	He will return now
LR,E = R < S	Geliyorum demiş	He has said he is coming	ST,S < E < R	Geleceğim diyecek	He will say he will come
ST,R < E < S	Geleceğim dedi	He said he will come	LR,S < E < R	Geleceğim diyecek	He will say he will come
LR,R < E < S	Geleceğim demiş	He has said he will come	ST,S = E < R	Geleceğim diyor	He is saying he will come
ST,R < S = E	Geldim diyor	He is saying that he came	LR,S = E < R	Geleceğim diyor	He is saying he will come
LR,R < S = E	Geldim diyor		ST,E < S < R	Yarın geleceğini söylediydi	He said that he will come tomorrow
ST,R < S < E	Geldim diyecek		LR,E < S < R	Yarın geleceğini söylemişti	He said that he will come tomorrow
LR,R < S < E	Geldim diyecek		ST,S < R = E	Geliyorum diyecek	He will say that he will come
ST,E < S = R	Şimdi döneceğini söylediydi		LR,S < R = E	Geliyorum diyecek	He will say that he will come
LR,E < S = R	Şimdi döneceğini söylemişti		ST, S < R < E	Geldim diyecek	He will say that he has come
ST,S = R = E	Geliyorum diyor		LR,S < R < E	Gelmişim diyecek	He will say that he has been coming

In Table 4, the LR preoperator indicates that the case is in the 'learning' state and ST indicates that the case is in the 'story' state. Similar to Table 2, Table 3 holds a translation to English column in order to give information about the meanings for non-Turkish readers, since most of the translations are not in exact but

close temporal form. Please note that in Table 4, there are some similar examples. These similar examples are not covered by Turkish since they are not possible by permutation. The extended version of Turkish temporal tenses applied over RTL is given in Table 4.

Table 4. Extended version of RTL for Turkish.

Relation	Sample	Relation	Sample
ST,E < R < S	Dün geleceğim dediydi	S = R = E	Geliyorum diyor
LR,E < R < S	*Dün geleceğim demişti*	S = R < E	Şimdiye dönecek
ST,E = R < S	Geliyorum dedi	S < E < R	Geleceğim diyecek
LR,E = R < S	*Geliyorum demiş*	S = E < R	Geleceğim diyor
ST,R < E < S	Geleceğim dedi	ST,E < S < R	Yarın geleceğini söylediydi
LR,R < E < S	*Geleceğim demiş*	LR,E < S < R	Yarın geleceğini söylemişti
R < S = E	Geldim diyor	S < R = E	Geliyorum diyecek
R < S < E	Geldim diyecek	ST,S < R < E	Geldim diyecek
ST,E < S = R	Şimdi döneceğini söylediydi	LR,S < R < E	Gelmişim diyecek
LR,E < S = R	Şimdi döneceğini söylemişti		

The original RTL in Table 3 holds 13 relations among the event, reference, and speech times. Table 4 adds 5 more relation types, the number extended to cover Turkish tenses. The italic forms in Table 4 indicate the newly added past tense, 'rivayet'.

The < MAKEINSTANCE > tag is modified and the alternatives mentioned below are added to make TimeML compatible with Turkish events:

```
tense::  = 'PAST'|'PRESENT'
|'FUTURE'|'NONE'|'INFINITIVE'
|'PRESPART'|'PASTPART'|'STORY'|'LEARNED'|'BEING'
aspect::  = 'PROGRESSIVE'|'PERFECTIVE'
|'PERFECTIVE_PROGRESSIVE'| 'NONE'
```

4.2. ATL and TimeML of Turkish temporal logic

The connection type of the TimeML in the link layer is inspired by ATL [42] but, unfortunately, does not support the full temporal logic for Turkish.

The link between events on the TimeML is represented using the 'TLink' tags. The Backus–Naur form of the TLink is quoted below:

```
lid ::  = LinkID
eventInstanceID ::  = EventInstanceID
timeID ::  = TimeID
signalID ::  = SignalID
relatedToEventInstance ::  = EventInstanceID
relatedToTime ::  = TimeID
relType ::  = 'BEFORE' | 'AFTER' | 'INCLUDES' | 'IS_INCLUDED' |
'DURING' | 'DURING_INV' | 'SIMULTANEOUS' | 'IAFTER' | 'IBEFORE' |
'IDENTITY' | 'BEGINS' | 'ENDS' | 'BEGUN_BY' | 'ENDED_BY'
```

Please note that the possibilities for the 'relType' symbol above are inspired by ATL and none of the above alternatives support the Turkish temporal case. To clarify the above alternatives, their explanations and examples are given in Table 5.

Table 5. ATL relation types.

relType	Comment	Example
BEFORE	An event finishes before another starts.	He came and saw the label.
AFTER	Reverse form of BEFORE and can substitute it.	
INCLUDES	A link to temporal expression or event that includes event.	John arrived in Boston last Thursday.
IS_INCLUDED	Reverse form of INCLUDES.	
DURING	A link to temporal expression or event while the event is in progress.	James was CTO for 2 years.
DURING_INV	Reverse form of DURING.	
SIMULTANEOUS	Two events at the same time.	
IAFTER	Immediately after	All passengers died when the plane crashed into the mountain.
IBEFORE	Reverse form of IAFTER.	
IDENTITY	Repeat of same verb.	He drove to Boston. During his drive he ate a donut.
BEGINS	A link to temporal expression or event that begins the event.	John was teaching since 1980.
ENDS	A link to temporal expression or event that ends the event.	John was at the gym until 7:00 pm.
BEGUN_BY	Reverse notation of BEGINS.	
ENDED_BY	Reverse notation of ENDS.	

Aside from the Reichenbach level, the ATL level is mostly similar to Turkish temporal logic. There are still some Turkish temporal differences from the TimeML. This section covers these differences, besides the missing parts common to all temporal logics but not covered in TimeML.

Unfortunately, ATL is not sufficient for a representation of Turkish temporal logic. The cases below state the temporal logic in Turkish where ATL is not sufficient. One of the specific problems for Turkish temporal logic is the positive/negative verb repetition. These terms in Turkish represent a continuous event by using 2 verbs with opposite meanings. For example, in English a single verb like 'blink' is represented in Turkish with 2 separate verbs, 'yanıp sönmek' (to light and to fade). This concept can also be represented by 'to flash' or 'to twinkle' in English – both single verbs. Another example is the translation of the term 'pacing up and down' or to 'pace back and forth' in Turkish. Again this term is represented by 2 separate verbs 'gelip gitmek' (to come and to go). Or another example: 'restart' in Turkish is 'kapatıp açmak' (to close and to open).

The above examples depict a group of terms in Turkish for which ATL is insufficient because it is not important to have a precise ordering of words in Turkish. For example, the term for 'restart' in Turkish, 'kapatıp açmak' (to close and to open), can also be represented as 'açıp kapatmak' (to open and to close), which is exactly the reverse order of the former term. On the other hand, the semantic representation of these terms is only 1 event in ATL, which creates a problem in the case of trying to represent a single event with 2 separate verbs.

ATL is a linear logic that is suitable for representing events in a linear manner. Unfortunately, the temporal logic behind Turkish is not exactly linear. Although there are some studies that model time in a nonlinear domain [39], the TimeML is implemented linearly using ATL. For example, we can try to represent

the Turkish sentence below in ATL. Some research about recursive events was carried out in [43], including the temporal prepositions, but these studies are limited for a single event recurring on the temporal basis. In the Turkish case, there may be multiple events following each other on the temporal basis.

"The life signal on the safety buoy was blinking while the divers were under water".

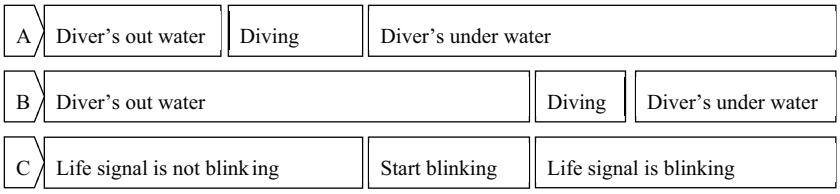

Figure 4. Sequence of events in sample sentence.

The above English sentence can easily be represented in ATL, as shown in Figure 4.

Since the starting time of the blinking of the life signal is unknown, either A-C or B-C or any time combination of "diving" and "start blinking" between these times is considered correct from the above input sentence.

The ATL representation of the above case would be as shown below.

Meets (SB, DUW) $\wedge$ During (DUW, LSB) $\wedge$ During (DOW, LSNB),

where "SB: signal blinking", "DUW: divers under water", "LSB: life signal blinking", "DOW: divers out water", and "LSNB: life signal is not blinking".

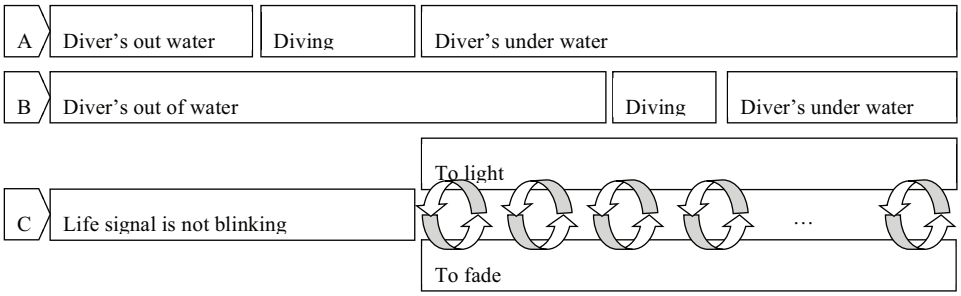

Figure 5. Suggestion for recurring events on ATL.

In the Turkish translation of the above sentence, the representation would be as shown in Figure 5.

The temporal logic behind Turkish states that even when the event starts with lighting or fading, these 2 events follow each other and continue while the diver is under the water. This logic cannot be stated in ATL. In many other languages, it is possible to state the same events using prepositions or modals instead of multiple events. The novel extension here is the recurrence between multiple events at a time.

Figure 6 visualizes all of the relations in ATL. The original figure [44], holding only the first 13 relations, does not cover the last row, which indicates that the relation type "recurs". This row is added to visualize one of the results of the study on ATL.

When RTL and ATL are separated, it is possible to see some potential integration problems. After solving the problems in the ATL and RTL levels, some problems still exist in Turkish and other temporal logics. This section deals with these kinds of unsolved problems that find answers neither in RTL nor in ATL.

One representation missing in the TimeML is that of self-referencing cases. This uncovered case in the TimeML is not unique to Turkish and the same problem can occur in any language. The speaker can refer to the current speech. For example, the case below is a self-referring case:

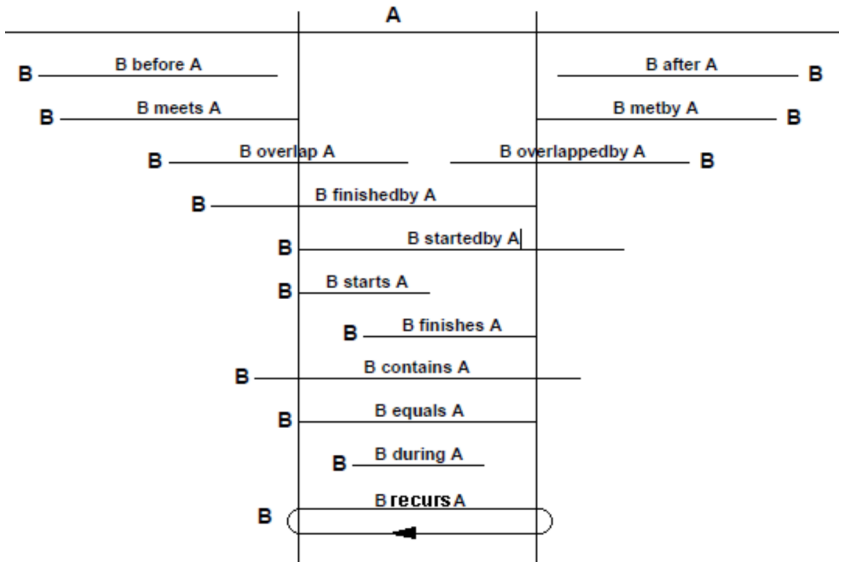

Figure 6. The addition of recurring events to ATL.

"My current talk is about computer science".

In the above sentence, the speaker refers to the current talk, which is the talk itself. This case creates a self-reference, which is not covered in ATL and therefore not in TimeML either. A solution would be to add this reference at the signal level of TimeML, but the signal level does not hold information regarding relationships between events. A better solution is suggested, which is adding these cases into the ATL level as a new relation type, as shown in Figure 7.

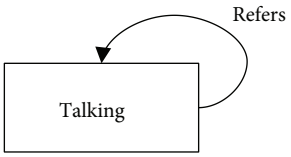

Figure 7. Self-referring events.

The self-reference problem, demonstrated in Figure 7, occurs in TimeML and this problem is neither a RTL nor ATL level problem, since RTL does not relate to the relation between 2 or more events and ATL does not relate to the time of reference.

Another piece missing from TimeML is the modeling of absence. At first glance, this concept could be created using the opposite of omnipresence. However, this does not translate exactly. The difference between the 2 is explained by the following examples.

"When do you play tennis?"

"Never".

The above dialogue is answered by the word "never", indicating that the event will never occur. Another example, with a slight semantic difference, would be:

"I never take notes when I attend classes".

This example is slightly different from the first example, because in the second case, the word "never" refers to the negativity of an event that is addressable in temporal logic.

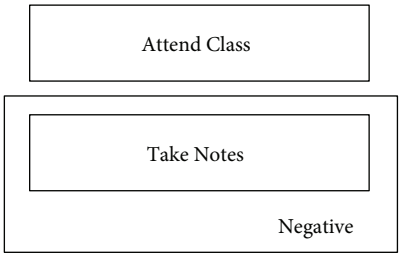

Figure 8. Relation of a negative event.

Figure 8 visualizes the relation between the events 'attending classes' and 'take notes', where the event 'take notes' is negative. The vertical lines indicate the state start and end.

This case can be interpreted as "I don't take notes when I attend classes". Thus, it can be interpreted that this event is connected to the 'attending of classes' by the 'during' operator with a negative perspective. On the other hand, the first example cannot be connected to any other event. Another problem for the first case is the demonstration of the event on a timeline. If an event never exists, the demonstration of the event is also impossible. The solution of adding nonexistence to the environment and connecting the event "never exist" to the nonexistence is suggested. This solution is also useful for the events connected by an order operator to nonexistence.

For example, the below sentence contains an event connected to nonexistence.

"Nothing existed before the Big Bang".

In this phrase, the word "nothing" indicates the nonexistence, and the event "Big Bang" occurs after "nothing", so it can be concluded that the Big Bang is connected to nonexistence by the "next" operator in temporal logic.

This solution can be demonstrated with a slightly more complicated example:

"Nothing comes from nothing" (Parmenides).

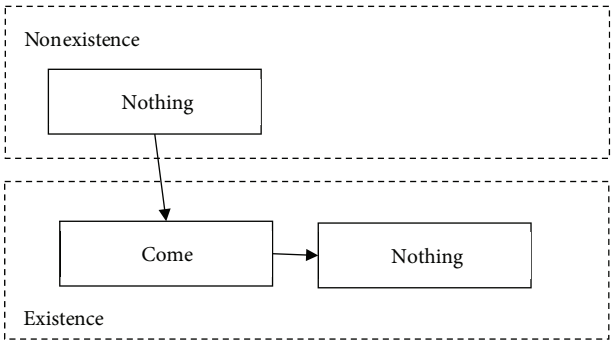

Figure 9. Interpretation of "nothing comes from nothing".

Figure 9 visualizes the famous quotation by Parmenides, where one of the "nothings" is in the existence domain and the other is in the nonexistence domain.

As a solution, TimeML should contain a nonexistence domain of events.

Another problem is the linearity of time. In Figure 1, when the linearity of temporal logic is explained, the linearity was considered as a forking of events in time. This concept is called multilinear time, where a timeline can flow through any of the possible paths.

5. Implementation and testing

This section covers the application of TimeML to Turkish texts with the modifications explained and recurrence.

Please note that this study is the first implementation of a temporal logic to Turkish, as there is no previous study or even a corpus covering Turkish.

In order to demonstrate that the above solutions are implementable in computer software, a new project was developed in Java with a user interface to model the temporal relations indicated above.

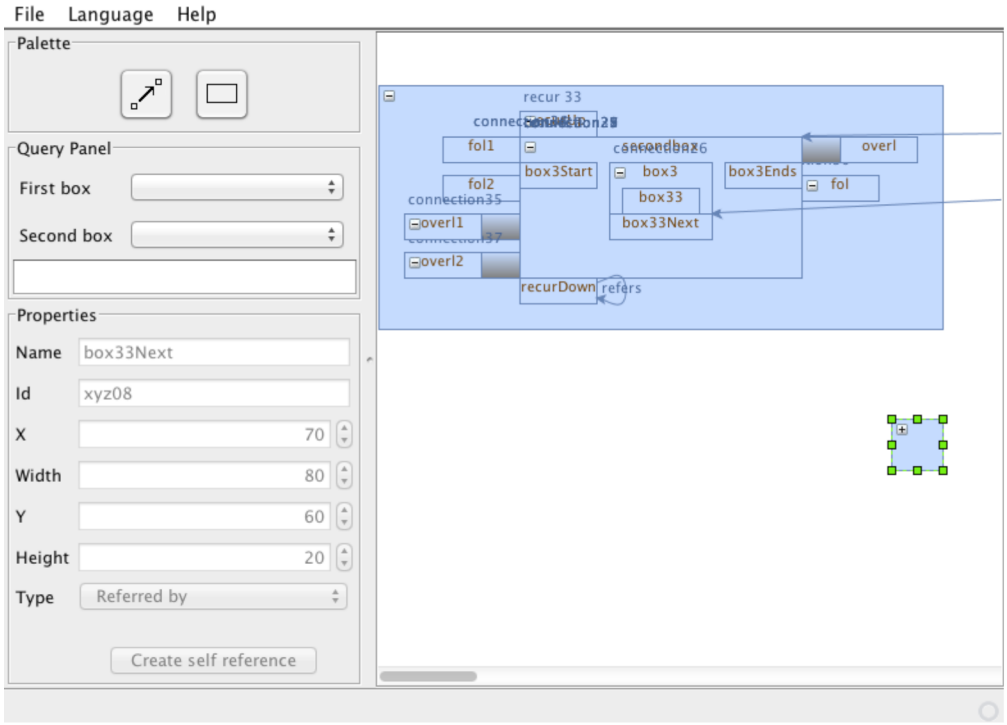

Figure 10. Screenshot of the software interface with a challenging problem.

The implemented temporal framework demonstrated in Figure 10 provides utilities to overcome all of the discussed problems above, like self-reference, negative events, or recurring events, as well as the current relation models of ATL or RTL. Using this new temporal framework, anyone can tag and create a visual demonstration of events in a Turkish text.

In order to keep the syntax structure simple, a corpus is garnered from 10 children's stories. Table 6 lists the numbers of sentences, words, and verbs in those stories. Each story in the corpus is numbered and the tables in this section hold the stories by numbers in the first column.

The categorization of the stories with respect to the tenses is given in Table 7.

Table 7 also underlines the number of instances of the learnt past tense, a tense that is very frequently used in stories. Table 7 also highlights the separation of past tense in Turkish. Table 8 lists the number of compound sentences that hold more than one event.

Table 8 also lists the number of compound sentences that hold more than one event. Table 9 displays the success of the original TimeML in the event level.

The tagged events are the events that can be tagged by the original TimeML rules. Untagged events are events with specific differences in Turkish that are not supported by the current TimeML. These increments, after the extension in Turkish, are displayed in Table 10.

Table 6. Number of sentences, words, and verbs in the corpus.

No.	No. of sentences	No. of words	No. of verbs
1	36	403	45
2	32	197	45
3	17	229	44
4	10	101	22
5	131	901	150
6	136	1217	155
7	109	898	160
8	43	372	65
9	31	365	55
10	35	259	45

Table 7. Tenses of the stories in the corpus.

#	Learnt past	Story past	Present	Future	Progressive
1	22	0	3	3	3
2	3	26	0	0	3
3	23	5	4	1	3
4	0	13	2	3	0
5	96	2	15	3	4
6	98	6	3	7	8
7	78	6	7	15	7
8	28	1	3	0	3
9	33	0	0	3	0
10	23	1	5	3	1

Table 8. Compound sentences in the corpus.

No.	Compound sentences	Percentage of compound sentences (%)
1	13	36
2	7	22
3	2	12
4	2	20
5	22	17
6	14	10
7	8	7
8	6	14
9	1	3
10	0	0

Since the extensions cover all of the possible cases in Turkish, success in the event level increases from 18% to 100% for the RTL level.

The extensions in ATL cannot be seen in the corpus study, since it is known these cases are possible in Turkish but none of the cases are found in the children's stories. These extended cases in ATL level are used very rarely in Turkish, although they are widely understood by speakers of Turkish.

In this study, TimeML has been enhanced a step beyond to cover Turkish temporal logic. After the above modifications, TimeML can be used in both Turkish and English and can more successfully model events.

Table 9. Preextention of the corpus in the event level.

No.	Tagged < EVENT >	Untagged < EVENT >	Success (%)	Failure (%)
1	9	57	14	86
2	3	20	13	87
3	8	31	21	79
4	5	5	50	50
5	22	144	13	87
6	18	138	12	88
7	29	114	20	80
8	6	55	10	90
9	3	49	6	94
10	9	31	23	78
Average			18%	82%
Sum	112	644		

Table 10. Extension results in the event level.

Story name	Tagged < EVENT >	Untagged < EVENT >	Success (%)	Story name	Tagged < EVENT >	Untagged < EVENT >	Success (%)
1	66	0	100	6	156	0	100
2	23	0	100	7	143	0	100
3	39	0	100	8	61	0	100
4	10	0	100	9	52	0	100
5	166	0	100	10	40	0	100

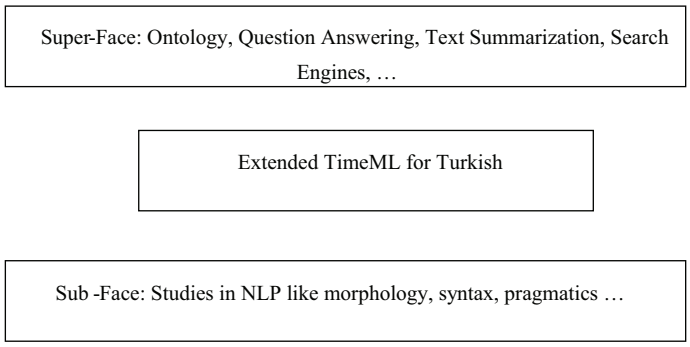

Figure 11. The 2 interfaces of the TimeML studies.

The overall view of this study and future studies can be demonstrated as in Figure 11. The extended version of TimeML has 2 interfaces. A future study can continue to apply this new version to ontology, question answering, text summarization, or any other study types. The newer version of TimeML also clears the path of temporal logics in Turkish. Research dealing with the ordering of events or temporal logics can use this new tool and may find a new focus as a result of and in response to this study.

6. Conclusion

This study was mainly aimed at the generation of a machine-computable temporal logic for Turkish. The literature review showed that there are currently many study types for different languages, and TimeML is the most mature and widely used machine-computable language for temporal statements.

Research into the extension of TimeML for Turkish temporal logic required the extension of the temporal logic behind TimeML to cover Turkish cases. In order to enhance TimeML, ATL and RTL were also enhanced. After the successful extension, the new version of TimeML was tested through a corpus and the results showed that the success of the extended TimeML for Turkish increased to 100%.

The success measured here is obviously corpus-dependent, and some cases may show a smaller increase in success or no increase at all. Unfortunately, we cannot compare the success with any other studies, since this is the first study in Turkish temporal reasoning. Moreover, the novel extensions on TimeML in this study were also tested on the TimeBank, which contains full English texts, and the success rate was not affected by the new additions.

The corpus studies also showed that there may be some cases in Turkish texts that are not covered by the formal linguistic studies on Turkish. The only case that we faced was due to translation from a foreign language. The translations may maintain the temporal logic of other languages, so for Turkish, there may still be a small number of cases that were not included in this study.

This study has 2 faces, the substructure and superstructure. People working on morphology, syntax, or semantics are also working on the subface of TimeML. TimeML declares a well-formed list of temporal rules that they can aim to gather during syntax studies, for example. On the superface, this study can be a base for further studies like question answering, search engines, temporal ontologies, or text summarization.

As a future direction in research, anybody can create a Turkish-to-TimeML processor by filling in the gaps between them, as morphological or syntactic studies have already been done in English. Any application built on TimeML for use in everyday life could also be a possibility for future work. For example, TimeML can be used for natural language translations; at least between Turkish and English, this study shows semantic differences that would be very useful in creating such an automatic translation. Creating a wider markup language to substitute for TimeML and cover all studies in other languages like Italian, Ukrainian, or French is another promising future direction for research.

Acknowledgment

This study was supported by the Scientific Research Projects Coordination Unit of İstanbul University under project numbers BYP-23343 and UDP-24502.

References

[1] Z. Orhan, I. Pehlivan, "Automatic knowledge extraction for filling in biography forms from Turkish texts", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 19, pp. 59–71, 2011.

[2] H. Sak, T. Güngör, Y. Safkan, "A corpus-based concatenative speech synthesis system for Turkish", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 14, pp. 209–223, 2006.

[3] I.D. El-Kahlout, K. Oflazer, "Exploiting morphology and local word reordering in English to Turkish phrase-based statistical machine translation", IEEE Transactions on Audio, Speech and Language Processing, Vol. 18, pp. 1313–1322, 2010.

[4] X. He, "Specifying and verifying real-time systems using time Petri nets and real-time temporal logic", Proceedings of the Sixth Annual Conference on Computer Assurance: Systems Integrity, Software Safety and Process Security, pp. 135–140, 1991.

[5] K.T. Narayana, A.A. Aaby, "Specification of real-time systems in real-time temporal interval logic", Proceedings of the 1988 Real-Time Systems Symposium, pp. 86–95, 1988.

[6] T.K. Shih, S.K.C. Lo, S.J. Fu, J.B. Chang, "Using interval temporal logic and inference rules for the automatic generation of multimedia presentations", Proceedings of the 3rd IEEE International Conference on Multimedia Computing and Systems, pp. 425–428, 1996.

[7] C. Dixon, M.C.F. Gago, M. Fisher, W. van der Hoek, "Using temporal logics of knowledge in the formal verification of security protocols", 11th International Symposium on Temporal Representation and Reasoning, pp. 148–151, 2004.

[8] R. Corin, A. Saptawijaya, "A logic for constraint-based security protocol analysis", IEEE Symposium on Security and Privacy, 2006.

[9] J. Luo, G. Luo, Y. Zhao, "Satisfiability degree computation for linear temporal logic", 10th IEEE International Conference on Cognitive Informatics & Cognitive Computing, pp. 373–380, 2011.

[10] M. Aiguier, C. Gaston, P. Le Gall, D. Longuet, A. Touil, "A temporal logic for input output symbolic transition systems", 12th Asia-Pacific Software Engineering Conference, pp. 15–17, 2005.

[11] S. Schewe, C. Tian, "Synthesising classic and interval temporal logic", 18th International Symposium on Temporal Representation and Reasoning, pp. 64–71, 2011.

[12] L. Deng, Y. Cai, C. Wang, Y. Jiang, "Fuzzy temporal logic on fuzzy temporal constraint networks", 6th International Conference on Fuzzy Systems and Knowledge Discovery. Vol. 6, pp. 272–276, 2009.

[13] J. Lv, J. Ying, M. Wu, T. Jiang, F. Zhu, "Verifying aspect-oriented programs using open temporal logic", 3rd IEEE International Conference on Secure Software Integration and Reliability Improvement, pp. 85–92, 2009.

[14] T. Caselli, F. dell'Orletta, I. Prodanof, "TETI: A TimeML compliant TimEx tagger for Italian", International Multiconference on Computer Science and Information Technology, pp. 185–192, 2009.

[15] A. Bittar, P. Amsili, P. Denis, L. Danlos, "French TimeBank: an ISO-TimeML annotated reference corpus", 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Vol. 2, pp. 130–134, 2011.

[16] J. Pustejovsky, "Time and the semantic Web", 12th International Symposium on Temporal Representation and Reasoning, pp. 5–8, 2005.

[17] C. Foriscu, "Why don't Romanians have a five o'clock tea, nor Halloween, but have a kind of Valentines day?", 9th International Conference on Computational Linguistics and Intelligent Text Processing, pp. 73–84, 2008.

[18] S. Im, H. You, H. Jang, S. Nam, H. Shin, "KTimeML: specification of temporal and event expressions in Korean text", Proceedings of the 7th Workshop on Asian Language Resources, pp. 115–122, 2007.

[19] D. Wonsever, A. Rosá, M. Malcuori, G. Moncecchi, A. Descoins, "Event annotation schemes and event recognition in Spanish texts", 13th International Conference on Computational Linguistics and Intelligent Text Processing, Vol. 2, pp. 206–218, 2012.

[20] W. Li, K.F. Wong, "A word-based approach for modeling and discovering temporal relations embedded in Chinese sentences", ACM Transactions on Asian Language Information Processing, Vol. 1, pp. 173–206, 2002.

[21] V. Rybakov, "Decidability w.r.t. logical consecutions of linear temporal logic extended by since and previous", Fundamenta Informaticae, Vol. 81, pp. 297–313, 2007.

[22] B. Boguraev, R.K. Ando, "TimeML-compliant text analysis for temporal reasoning", 19th International Joint Conference on Artificial Intelligence, pp. 997–1003, 2005.

[23] F. Schilder, G. Katz, J. Pustejovsky, "Annotating, extracting and reasoning about time and events", International Conference on Annotating, Extracting and Reasoning About Time and Events, pp. 1–6, 2005.

[24] E. Saquete, "TERSEO + T2T3 Transducer: A systems for recognizing and normalizing TIMEX3", 5th International Workshop on Semantic Evaluation, pp. 317–320, 2010.

[25] P. Muller, A. Reymonet, "Using inference for evaluating models of temporal discourse", 12th International Symposium on Temporal Representation and Reasoning, pp. 11–19, 2005.

[26] W.E. Hinrichs, "Tense, quantifiers, and contexts", Computational Linguistics, 14, pp. 3–14, 1988.

[27] W.E. Hinrichs, "A compositional semantics of temporal expressions in English", 25th Annual Meeting on Association for Computational Linguistics, pp. 8–15, 1987.

[28] L. Zhou, G. Hripcsak, "Methodological review: temporal reasoning with medical data–a review with emphasis on medical natural language processing", Journal of Biomedical Informatics, Vol. 40, pp. 183–202, 2007.

[29] I. Androutsopoulos, G. Ritchie, P. Thanisch, "Time, tense and aspect in natural language database interfaces", Natural Language Engineering, Vol. 4, pp. 229–276, 1998.

[30] K. Hacioglu, Y. Chen, B. Douglas, "Automatic time expression labeling for English and Chinese text", Proceedings of the Sixth International Conference on Computational Linguistics and Intelligent Text Processing, pp. 548–559, 2005.

[31] I. Mani, G. Wilson, "Robust temporal processing of news", Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pp. 69–76, 2000.

[32] B.B. Rieger, "On understanding understanding. Perception-based processing of NL texts in SCIP systems, or meaning constitution as visualized learning", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 34, pp. 425–438, 2004.

[33] B. Shen, Z. Zhang, C. Yuan, "Person name identification in Chinese documents using finite state automata", IEEE/WIC International Conference on Intelligent Agent Technology, pp. 478–481, 2003.

[34] Y. Zhou, X. Lin, "Mechanism of school organizational knowledge development", Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering, pp. 818–824, 2005.

[35] S. Park, J.K. Aggarwal, "Semantic-level understanding of human actions and interactions using event hierarchy", 2004 Conference on Computer Vision and Pattern Recognition Workshop, p. 12, 2004.

[36] G. Ro, S. Bensalem, "Allen linear (interval) temporal logic – translation to LTL and monitor synthesis", Proceedings of the 18th International Conference on Computer Aided Verification, pp. 263–277, 2006.

[37] H. Reichenbach, Elements of Symbolic Logic, New York, NY, USA, Macmillan, 1947.

[38] A. Özkırımlı, Türk Dili: Dil ve Anlatım, İstanbul, Turkey Bilgi University Press, 2001 (in Turkish).

[39] D.F. Anger, D. Mitra, V.R. Rodriguez, "Satisfiability in nonlinear time: algorithms and complexity", Proceedings of the 12th International FLAIRS Conference, pp. 406–411, 1999.

[40] J. Pustejovsky, J. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, G. Katz, "TimeML: robust specification of event and temporal expressions in text", New Directions in Question Answering, pp. 28–34, 2003.

[41] N. Kotsyba, "Using petri nets for temporal information visualization", Études Cognitives/Studia Kognitywne, Vol. 7, pp. 189–207, 2006.

[42] R. Saurí, J. Littman, B. Knippen, R. Gaizauskas, A. Setzer, J. Pustejovsky, TimeML Annotation Guidelines Version 1.2.1, 2006, available at http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf.

[43] I. Pratt-Hartmann, "Temporal prepositions and their logic", Artificial Intelligence, Vol. 166, pp. 1–36, 2006.

[44] P. Bellini, R. Mattolini, P. Nesi, "Temporal logics for real-time system specification" ACM Computing Surveys (CSUR), Vol. 32, pp. 12–42, 2000.