International Journal of Semantic Computing

World Scientific
www.worldscientific.com

EVENT AND TEMPORAL EXPRESSION EXTRACTION FROM RAW TEXT: FIRST STEP TOWARDS A TEMPORALLY AWARE SYSTEM

NAUSHAD UZZAMAN

Department of Computer Science
University of Rochester, Rochester, NY, USA
naushad@cs.rochester.edu
http://www.cs.rochester.edu/~naushad

JAMES F. ALLEN

Department of Computer Science
University of Rochester, Rochester, NY, USA
james@cs.rochester.edu
http://www.cs.rochester.edu/~james

Received (08 November 2010)
Revised (09 January 2011)
Accepted (15 January 2011)

Extracting temporal information from raw text is fundamental for deep language understanding, and key to many applications like question answering, information extraction, and document summarization. Our long-term goal is to build complete temporal structure of documents and use the temporal structure in other applications like textual entailment, question answering, visualization, or others. In this paper, we present a first step, a system for extracting events, event features, main events, temporal expressions and their normalized values from raw text. Our system is a combination of deep semantic parsing with extraction rules, Markov Logic Network classifiers and Conditional Random Field classifiers. To compare with existing systems, we evaluated our system on the TempEval 1 and TempEval 2 corpus. Our system outperforms or performs competitively with existing systems that evaluate on the TimeBank, TempEval 1 and TempEval 2 corpus and our performance is very close to inter-annotator agreement of the TimeBank annotators.

Keywords: Event extraction; temporal expression extraction; main event identification, TimeBank; TempEval; temporal information processing; information extraction; TRIPS; TRIOS.

1. Introduction

The recent emergence of language processing applications like question answering, information extraction, and document summarization has drawn attention to the need for systems that are temporally aware. For example, for a QA system in newswire domain, if we want to know who was the Prime Minister (PM) of Bangladesh in the February of 1995, and we only had documents that tell us about the PM from 1994 to 1999 then a temporally aware system will help the QA system to infer who was PM in the February of 1995 as well. In medical domain for patient's history record, doctors write all the information about patients' medical record, usually not in chronological order. Extracting a temporal structure of the medical record will help practitioner understand the patient's

medical history easily. For people who have trouble reading and understanding, be it people with cognitive disabilities or non-native English speakers, a temporal structure of document could help them to follow a story better. Extracting temporal information will benefit in almost any application processing natural language text.

In a temporally aware system, eventualities and temporal expressions are fundamental entities. In this paper, we present our work on extracting both of these from raw text.

To explain our system for extracting temporal entities, we start with describing related efforts. Then, we describe our system to extract temporal entities. Next, we show the performance of our system and compare with other systems on TimeBank [1] (or TempEval-1 [2]) and TempEval-2 [3] corpus. Finally, we conclude by describing our future directions.

2. Related Work

To a first approximation, our view of events matches closely with the TimeML [4] temporal annotation scheme. They consider *events* a cover term for situations that *happen* or *occur*. Events can be punctual or last for a period of time. They include predicates describing *states* or *circumstances* in which something obtains or holds true.

A few existing systems have been tested on the TimeBank. Sauri et al. [5] implemented an event and event feature extraction system EVITA and showed that a linguistically motivated rule-based system, with some statistical guidance for disambiguation, could perform well on this task. Bethard and Martin [6] applied statistical machine learning algorithms on the Timebank corpus to build models for event extraction and event *class* classification. Chambers et al. [7] assumed events from TimeBank and extracted event features using machine-learning algorithms. There are also a few other notable recent systems from TempEval-2[a] [3] that approached event extraction as well. Hector et al.'s [8] TIPSem system additionally considers semantic role labels using a Conditional Random Field classifier. Grover et al.'s [9] Edinburgh-LTG system initially use their IE system based on LT-XML2[b] and LT-TTT2[c] toolkit. Then they convert their IE system output to TimeML events and event features using hand coded rules.

There also has been other work on event extraction on other corpora. One significant effort includes systems like Ahn [10], Aone and Ramos-Santacruz [11] and many others, which are based on or related to the MUC-7[d] or ACE[e] specifications, which specify the task as not just extracting the event but also extracting event arguments, assigning roles and determining event coreference. But in these cases the entities are limited and pre-

[a] TempEval is also based on TimeML
[b] http://www.ltg.ed.ac.uk/software/ltxml2
[c] http://www.ltg.ed.ac.uk/software/lt-ttt2
[d] Message Understanding Conference
[e] http://www.nist.gov/speech/tests/ace/

defined set like person, organization, location, geo-political entity, facility, vehicle, weapon, etc. We are interested in all events like TimeML, instead of a limited set of entities with detailed information.

Another line of significant work is event extraction in Biological domains, especially the BioNLP shared task on event extraction [12]. Their event extraction is defined based on the biological domain dependent GENIA[f] ontology. Their shared task include subtasks like, i) core-event detection with the primary argument proteins, ii) event enrichment by extracting secondary arguments and iii) detection of negation and speculation statements concerning extracted events. We don't compare our system with systems from BioNLP, because their event extraction evaluation was based on domain specific ontology and we are only interested in natural language text.

The main event in TimeML identifies the most important event in the sentence. The main event is encoded in TimeML complaint corpora by a TLINK (temporal link) between the main events of consecutive sentences. Identifying main events is the first step to identify which pair of events to consider classifying temporal relations between events in consecutive sentences. An automatic summarization system can also take benefit of main events by considering it as the main idea of the sentence [25]. In our knowledge, there is no other work that has approached this task of main event identification. Part of reason could be that it was never a TempEval task.

Finally, a large number of systems that extract temporal expressions were developed in the scope of the ACE Temporal Expression Recognition and Normalization[g] (TERN), in which TIMEX2 tags are associated with temporal expressions. There are few differences between TimeML TIMEX3 and TERN TIMEX2, notably TIMEX2 includes post-modifiers (prepositional phrases and dependent clauses) but TIMEX3 doesn't. But to a large extent TIMEX3 is based on TIMEX2. Boguraev and Ando [13] and Kolomiyets and Moens [14] reported performance on recognition of temporal expressions using TimeBank as an annotated corpus. Boguraev and Ando's work is based on a cascaded finite-state grammar (500 stages and 15000 transitions) and Kolomiyets and Moens first filter certain phrase types and grammatical categories as candidates for temporal expressions and then apply Maximum Entropy classifiers. Ahn et al. [15], Hachioglu et al. [16] and Poveda et al. [17] used approaches with a token-by-token classification for temporal expressions represented by a B-I-O encoding[h] with lexical and syntactic features and tested on the TERN dataset. Recently, Strotgen et al. [18] used a rule based technique for recognition and normalization of temporal expressions in TempEval-2.

[f] http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/genia-ontology.html
[g] http://fofoca.mitre.org/tern.html
[h] In B-I-O encoding, B stands for "beginning of the sequence", I stands for "inside the sequence" and O stands for "outside the sequence".

We extract both events and temporal expressions. Hence, we evaluate our system for both events and temporal expression extraction together on temporally annotated TimeBank [1] (or TempEval-1 [2]) and TempEval-2 [3] corpora.

3. Our System Modules

Our approach for event extraction is linguistically motivated. We do deep semantic parsing and then extract events and features using hand-build rules that do graph matching on the logical forms. We then use a Markov Logic Network (MLN) classifier to filter out errors. We also implemented MLN classifiers to directly classify main event and event features *class, tense, aspect* and *pos* from surface features extracted from the text and derived from TRIPS parser output. For temporal expressions, we use both TRIPS parser's extraction and a Conditional Random Field based classifier. Our overall framework is shown in Figure 1.

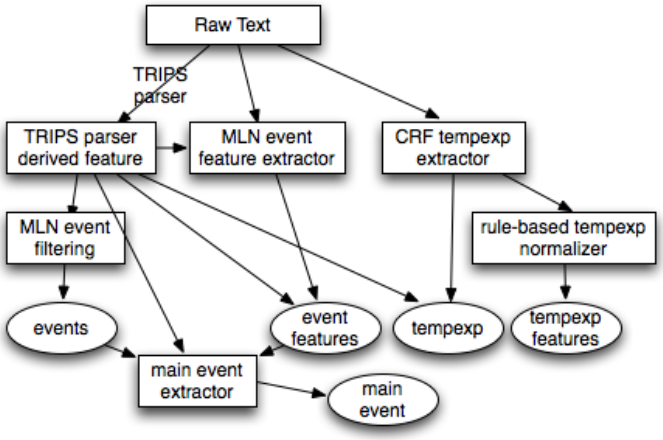

Fig 1. Framework of the system

Before describing these techniques, we will describe two significant modules of our system, the TRIPS parser and the Markov Logic Network classifier.

3.1. *TRIPS Parser*

We use the TRIPS parser (Allen et al) [19] to produce deep logical forms of text. The TRIPS grammar is lexicalized context-free grammar, augmented with feature structures and feature unification. The grammar is motivated from X-bar theory, and draws on principles from GPSG (e.g., head and foot features) and HPSG. The parser uses a packed-forest chart representation and builds constituents bottom-up using a best-first search strategy similar to A*, based on rule and lexical weights and the influences of the statistical preprocessing. The search terminates when a pre-specified number of spanning constituents have been found or a pre-specified maximum chart size is reached. The chart

is then searched using a dynamic programming algorithm to find the least cost sequence of logical forms according to a cost table that can be varied by genre.

The TRIPS system uses a range of statistically driven preprocessing techniques, including part of speech tagging, constituent bracketing, interpretation of unknown words using WordNet, and named-entity recognition. All these are generic off-the-shelf resources that extend and help guide the deep parsing process.

The TRIPS LF (logical form) ontology[i] is designed to be linguistically motivated and domain independent. The semantic types and selectional restrictions are driven by linguistic considerations rather than requirements from reasoning components in the system [20]. Word senses are defined based on subcategorization patterns and domain independent selectional restrictions. As much as possible the semantic types in the LF ontology are compatible with types found in FrameNet [21]. FrameNet generally provides a good level of abstraction for applications since the frames are derived from corpus examples and can be reliably distinguished by human annotators. However TRIPS parser uses a smaller, more general set of semantic roles for linking the syntactic and semantic arguments rather than FrameNet's extensive set of specialized frame elements. The LF ontology defines approximately 2500 semantic types and 30 semantic roles. The TRIPS parser will produce LF representations in terms of this ontology.

A very simple example, the result of parsing the sentence, *He fought in the war*, is expressed as the following set of expressions in an unscoped logical formalism with reified events and semantic roles.

```
(SPEECHACT V1 SA-TELL :CONTENT V2)
(F V2 (:* FIGHTING FIGHT) :AGENT V3 :MODS (V4) :TMA ((TENSE PAST)))
(PRO V3 (:* PERSON HE) :CONTEXT-REL HE)
(F V4 (:* SITUATED-IN IN) :OF V2 :VAL V5)
(THE V5 (:* ACTION WAR))
```

The main event `(V2)` is an event of type `FIGHTING`, which is a subclass of `INTENTIONAL-ACTION`, and which corresponds to the first WordNet sense of fight, and includes verbs such as *fight, defend, contend* and *struggle*. The `:AGENT` role of this event is the referent of the pronoun "he", and the event is `SITUATED-IN` an event described by the word "war". For words not in the TRIPS core lexicon, the system looks up the WordNet senses and maps them to the TRIPS ontology. The word *war* is not in the core lexicon, and via WordNet is classified into the TRIPS ontology as the abstract type `ACTION`.

[i] TRIPS ontology browser: http://www.cs.rochester.edu/ research/trips/lexicon/browse-ont-lex.html

3.2. *Markov Logic Network (MLN)*

One of the statistical relational learning (SRL) frameworks that recently gained attention as a platform for global learning and inference in AI is Markov Logic [22]. Markov logic is a combination of first order logic and Markov networks. It can be viewed as a formalism that extends first-order logic to allow formulae to be violated with some penalty. Formally, an MLN is a set of weighted first-order formulae. Given a set of constants, an MLN can be instantiated into a ground Markov network where each node is an atom. Each formula represents a feature in the grounded Markov network with the corresponding weight. The probability of an assignment x is $P(x) = 1/Z * exp(\sum_i w_i n_i(x))$, where Z is normalization constant, w_i is the weight of the i^{th} formula and $n_i(x)$ is the number of satisfied groundings for the i^{th} formula.

For our different classification tasks, we used different classifiers based on MLN. We used an off-the-shelf MLN classifier *Markov thebeast*[j], using Cutting Plane Inference [23] with an Integer Linear Programming (ILP) solver for inference.

To use *thebeast* or any other MLN framework, at first we have to write the formulas, which is equivalent to defining features for other machine learning algorithms. The Markov network will learn the weights for these formulas from the training corpus and use these weights for inference in testing phase.

One easy example will give a brief idea about these weights. To classify the event feature *class*, we have a formula that captures influence of both *tense* and *aspect* together. Here are three examples that show the learnt weights for the formula from training data.

```
tense(e1, INFINITIVE) & aspect(e1, NONE)
                        => class(e1, OCCURRENCE) weight = 0.319913

tense(e1, NONE) & aspect(e1, NONE)
                           => class(e1, STATE) weight = 0. 293119

tense(e1, PRESPART) & aspect(e1, NONE)
                        => class(e1, REPORTING) weight = -0.268185
```

The MLN then uses these weights for making inference about the *class*. Generally, larger the weights are, the more likely the formula holds. These weights could be negative as well, i.e. the formulas are most likely not to hold. For example, the INFINITIVE form (e.g. *change* in "It is not going to change") is coded as an OCCURRENCE most of the time in the corpus; hence the weight for that formula (which is also automatically induced by MLN) is higher.

Finding useful features for MLNs is the same as any other machine learning algorithms. However, the MLN framework gives the opportunity to combine different features in first order logic, which can lead to better inference. For example, when filtering events, we have formula combining *word and pos*, or *word and previous word*,

[j] http://code.google.com/p/thebeast/

or *pos and next pos*, where we can capture relationship of two predicates together. Many of these predicates (features) could be encoded in other classifiers by concatenating the features, but as the size of a formula increases it complicates matters and we have to regenerate the whole classifier data, every time we introduce a new relationship.

4. Event and Event Feature Extraction

For event extraction, we parse the raw text with the TRIPS parser. Then we take the resulting Logical Form (LF) and apply around hundred of hand-coded extraction patterns to extract events and features, by matching semantic patterns of phrases. These hand-coded rules are devised by checking the parse output in our development set. It was 2-3 weeks of work to come up with most of the extraction rules that extract the events. There were only minor incremental improvements in rules afterwards. It is worth mentioning that these rules are very generic and can be used in new domains without any extra work because the TRIPS parser and ontology are domain independent, and use mappings from WordNet to interpret unknown words. Hence, the extraction rules will apply (and can be tested) for any natural language text without any extra work.

Because of the ontology, we can usually express general rules that capture a wide range of phenomena. For instance, all noun-phrases describing objects that fall under the TRIPS Ontology's top-level type `situation-root` are extracted as described events. This situation is captured by a single extraction rule:

```
((THE ?x (? type SITUATION-ROOT))
   -extract-noms>
   (EVENT ?x (? type SITUATION-ROOT):pos NOUN :class OCCURRENCE ))
```

Since *war* has the type *action*, which falls under `situation-root` in TRIPS ontology, this extraction rule will match the LF `(THE V5 (:* ACTION WAR))` and will extract *war* as event. Beside matching *war* under `situation-root` in ontology, it also matches the specifier *the*, which indicates that it is a definite noun phrase.

The result of matching around hundred of such rules to the sentence "He fought in the war", will extract events as follows:

```
<EVENT eid=V2 word=FIGHT pos=VERBAL ont-type=FIGHTING
     class=OCCURRENCE tense=PAST voice=ACTIVE aspect=NONE
     polarity=POSITIVE>
<RLINK eventInstanceID=V2 ref-word=HE ref-ont-type=PERSON
     relType=AGENT>
<SLINK signal=IN eventInstanceID=V2 subordinatedEventInstance=V5
     relType=SITUATED-IN>
<EVENT eid=V5 word=WAR pos=NOUN ont-type=ACTION class=OCCURRENCE
     voice=ACTIVE polarity=POSITIVE aspect=NONE tense=NONE>
```

In this way, we extract events and TimeML-suggested event features (*class, tense, aspect, pos, polarity, modality)*. We also extract a few additional features, shown in boldface, such as ontology type (ont-type). TimeML tries to capture event information by very high-level *class* or *pos*. The ontology type feature captures more fine-grained information about the event, but still much higher level than the words. The extraction rules also map our fine-grained ontology types to the coarse-grained TimeML event class. We also extract relations between events (SLINKs) whenever one event syntactically dominates the other, so it extracts more than TimeML's SLINKs, as well as another new relation between events and its arguments (RLINK). Details about these new additions can be found in UzZaman and Allen [24].

The TRIPS parser extracts events from the TimeBank corpus (and also TempEval 1 and 2) with very high recall compared to any other existing systems. However, this high performance comes with the expense of precision. The reasons for low precision include, 1) the fact that generic events are not coded as events in TimeML scheme (detail in Evaluation and Discussion section), 2) errors of TRIPS parser and, 3) legitimate events according to TimeML scheme that are found by the parser but missed by TimeBank annotators. To remedy some of these problems, we introduced a MLN based filtering classifier, using the event features extracted from TRIPS parser. As features, we used the following features and also some of their combinations in MLN first order logic formulas - lexical features: word, stem, position in sentence, next word, previous word, previous word of verbal word sequence[k], contains dollar, suffix; syntactic features: part-of-speech tag, tense, voice, polarity, TimeML aspect, modality, verbal pos sequence, previous pos of verbal pos sequence, next pos, previous pos and semantic features: abstract semantic class - ontology type, TimeML class, semantic roles and their arguments of events. The syntactic and semantic features are generated from TRIPS parser output. There are two goals for this filtering step:

- *Eliminating events that result from errors in the parse,*
- *Removing some event-classes, such as generics, those were not coded in TimeML, for evaluation on TimeML compliant corpora.*

As noted, the second goal is needed to perform a meaningful evaluation on the TimeML compliant corpora. The resulting system, including parsing, extraction, and post-filtering, is named as TRIOS system.

The parser and extraction rules already give us event features along with events. But due to the limitation of the parser in newswire domain in general, we are still not outperforming other existing systems on event features. Instead of using parser-extracted features, we implemented MLN classifiers to classify the *class, tense, aspect* and *pos* features, using the features generated from the TRIPS parser plus lexical and syntactic

[k] *Verb word sequence* is a penn tag derived features, which captures all previous verbs, or TO (infinitive), or modal verbs, of the event word. That is, it will capture all consecutive verbs before the event until we get non-verbal word. Similarly *verb pos sequence* is the penn tag sequence of these verbs.

features generated from the text using the Stanford POS tagger. Table 1 gives a summary of attributes used to classify these event features.

Table 1. Attributes/features used for classifying event features *pos, tense, aspect* and *class*

Event feature	**Common attributes / features used in MLN classifiers**	**Extra attributes used in MLN classifiers**
pos	event word, event penn tag, verb pos sequence, verb word sequence, previous word of verb sequence, previous pos of verb sequence, next word, next pos	TRIPS pos suggestions
tense		TRIPS tense suggestions, pos, polarity, modality, voice (active or passive)
aspect		TRIPS aspect suggestions, pos, polarity, modality, voice (active or passive), pos x previous pos, pos x next pos
class		TRIPS class suggestions, ont-type, tense x aspect, pos, stem, contains dollar

5. Main Event Identification

The main event represents the most important event in the sentence, i.e. it represents the main idea or main concept of the sentence. An example will help to understand the main event better. For the sentence, *"Also today, King Hussein of Jordan **arrived** in Washington seeking to mediate the Persian Gulf crisis."*, we have four events, *arrived, seeking, mediate* and *crisis*. But *arrived* is the main event here. To build a temporally aware system we need to identify the temporal relations between main events of the consecutive sentences. On the other hand, main events can also be used for sentence level summarization [25].

We approach the problem of identifying main events, given all the events, as another classification problem. We take our extracted events from the previous step and run a MLN classifier to classify the main events of the sentences. As features, we used the following features and also some of their combinations in MLN first order logic formulas - lexical features: word, stem, position in sentence[l], next word, previous word, previous word of verbal word sequence[m], contains dollar, suffix; syntactic features: part-of-speech tag, tense, voice, polarity, TimeML aspect, modality, verbal pos sequence, previous pos of verbal pos sequence, next pos, previous pos and semantic features: abstract semantic class - ontology type, TimeML class, semantic role of event, abstract semantic class (ont type) of event's argument, if event has a temporal expression as argument.

The semantic features are generated from TRIPS parser output. A few syntactic features are generated from TRIPS parser and *class, tense, aspect* and *pos* are generated

[l] We split the sentence in four segments and "position in sentence" captures in which segment the event belong, i.e. if first segment, or second segment, or third segment or last segment.

[m] *Verb word sequence* is a penn tag derived features, which captures all previous verbs, or TO (infinitive), or modal verbs, of the event word. That is, it will capture all consecutive verbs before the event until we get non-verbal word. Similarly *verb pos sequence* is the penn tag sequence of these verbs.

using MLN classifier (check Table 1 and previous section for details) and Penn part-of-speech tag is generated using Stanford POS tagger.

As a back-off model, i.e. if our system fails to identify main event in a sentence, we use the first verbal event as main event. If no verbal event exists in the sentence then we consider the first event as the main event. We show in the Evaluation that this model is a good baseline and adding this back-off model improves the performance significantly as well. We also show that with very naïve features such as the lexical features, the part-of-speech tag derived features and the back-off model, we can get a high performing system, which performs better than a classifier trained on the TimeML-defined event features. However, we get the best performance by including the semantic features.

In the TempEval-2 data we have found that approximately 1.5% of the sentences have more than one main event. This might not be a very significant number, but these instances are important and significant in many domains. They represent conjunctions or colon separated sentences. Our framework handles these sentences because we don't have any constraint that there could be at most one main event in a sentence. Our classifier considers features like semantic information, semantic roles and other features. If any event has features related to main events then the classifier will try to classify it as main event. As a result, we handle the conjunctions and colon-separated sentences without any extra work.

6. Temporal Expression Extraction

6.1. *Recognizing Temporal Expression*

Our temporal expression extraction module is a hybrid between traditional machine learning classifier and the TRIPS parser extractor. For the machine learning classifier, we used a token-by-token classification for temporal expressions represented by a B-I-O[n] encoding with a set of mostly lexical features, using Conditional Random Field (CRF) classifier[o]. Specifically, we use as features the word, and a set of features indicating whether the word describes a year, month, number, time-string, day of the week, temporal expressions modifier[p], shape[q] of word, etc. We then use CRF++ templates to capture the relation between the different features in a sequence to extract the temporal expressions. For example, we will write a template to capture that the current word is a

[n] In B-I-O encoding, B stands for "beginning of the sequence", I stands for "inside the sequence" and O stands for "outside the sequence".

[o] We used off the shelf CRF++ implementation. http://crfpp.sourceforge.net/

[p] Few temporal expression modifiers are: *this, mid, first, almost, last, next, early, recent, earlier, beginning, nearly, few, following, several, around, the, less, than, more, no, of, each, late.*

[q] *shape* is also named as *pattern* or *short-types* in the literature. Shape is defined by using the symbols X, x and 9. X and x are used for character type as well as for character type patterns for representing capital and lower-case letters for a token. 9 is used for representing numeric tokens. For example, March 1st will have the shape "Xxxxx 9xx".

modifier and the next word is a *time-string*, this rule will train the model to capture sequences like *this weekend*, *earlier morning*, *several years*, etc.

Separately, TRIPS parser extracts temporal expressions the same way as we extract events. The performance of TRIPS parser's temporal extraction alone doesn't outperform state-of-the-art techniques on the evaluation measures. However, TRIPS does extract some temporal expressions that are missed by our CRF based system and even sometimes missed by TimeBank annotators. So we implemented a hybrid system using both TRIPS and the CRF-based system. The temporal expressions suggested by the TRIPS parser but missed by the CRF-based system are passed to a filtering step that tries to extract a normalized value and type of the temporal expression (see next section). If we can find a normalized value and type, we accept these temporal expressions along with CRF based system's extracted temporal expressions.

6.2. *Determining The Normalized Value and Type of Temporal Expressions*

Temporal expressions are most useful for later processing when a normalized *value* and *type* is determined. The task of determining normalized value and type was also a subtask for 2010 TempEval challenge[r].

Table 2. Examples of normalized *values* and *types* for temporal expressions according to TimeML

Temporal exp.	***Type***	***Value***
DCT (given): March 1, 1998; 14:11 hours	TIME	1998-03-01T14:11:00
Sunday	DATE	1998-03-01
last week	DATE	1998-W08
mid afternoon	TIME	1998-03-01TAF
nearly two years	DURATION	P2Y
each month	SET	P1M

We implemented a rule-based technique to determine the *types* and *values*. We match regular expressions to identify the *type* of temporal expressions. *Type* could be either of *TIME, DATE, DURATION and SET*. We then extracted the normalized value of temporal expression, as defined by the TimeML scheme. We took the Document Creation Time (DCT) from the documents and then calculated the values for different dates, e.g. last month, Sunday, today. Table 2 shows some temporal expressions examples with normalized *value* and *type*. Our temporal expression normalizer is available[s] for public use.

7. Evaluation and Discussion

[r] http://www.timeml.org/tempeval2/
[s] Available online at: http://www.cs.rochester.edu/u/naushad/temporal

7.1. *Event Extraction*

As mentioned before, the TimeBank corpus [1] is annotated according to TimeML specification. For TempEval-1 (Temporal Evaluation contest) [2], the same corpus was released with modified event relations and event features. Before describing our results and comparing with others, it is important to more carefully define the notion of *event* according to the TimeML specification.

As mentioned earlier, TimeML considers *events* to be a cover term for situations that *happen* or *occur*. Events can be punctual or last for a period of time. They consider predicates describing *states* or *circumstances* in which something obtains or holds true. Events are generally expressed by means of tensed or untensed verbs, nominalizations, adjectives, predicative clauses, or prepositional phrases. In addition, the TimeML specification says not to tag generic interpretations, even though capturing them could be of use in question answering. By generics, they mean events that are not positioned in time or in relation to other temporally located events in the document. For example, they won't annotate *use* and *travel* in the sentence: *Use of corporate jets for political travel is legal.*

In addition, subordinate verbs that express events which are clearly temporally located, but whose complements are generics, are not tagged, For example, *He said participants are prohibited from mocking one another*. Even though the verb *said* is temporally located, it isn't tagged because its complement, *participants are prohibited from mocking one another*, is generic. While we feel such events are important to temporally locate, for the evaluation we stay with the TimeML specification.

And finally, event nominalizations that don't provide any extra information beyond the verb that dominates it are also not tagged. For example, in "Newspaper reports have **said** ...", only *said* is annotated and *reports* won't be annotated.

As for event attributes, TimeML considers *class, tense, aspect, and nf_morph (Non-finite morphology)*. They use only seven abstract event classes rather than more fine-grained classes in the TRIPS ontology: (1) Occurrence: e.g., die, crash, build; (2) State: e.g., on board, kidnapped; (3) Reporting: e.g., say, report; (4) I-Action: e.g., attempt, try, promise; (5) I-State: e.g., believe, intend, want; (6) Aspectual: e.g., begin, stop, continue; and (7) Perception: e.g., see, hear, watch, feel.

For *tense*, they use *PAST, PRESENT, FUTURE, NONE*; for *aspect*, they use *PROGRESSIVE, PERFECTIVE, PERFECTIVE_PROGRESSIVE, NONE;* and for nf_morph, they consider *ADJECTIVE, NOUN, PRESPART, PASTPART, INFINITIVE.*

TimeBank contains with around 200 newswire documents. Later in the TempEval contest, they used the same documents of TimeBank with some modification. One modification involved removing the *nf_morph* attribute and introducing *pos* tag (part of speech) with *VERB, ADJECTIVE, NOUN, PREPOSITION, OTHER*. They modified the tense with *PRESENT, NONE, PAST, FUTURE, INFINITIVE, PRESPART, PASTPART*, to include rest of the values of *nf_morph*.

Our first experiments are on TempEval-1 corpus. As a result, none of the systems we compare to contain performance on the *pos* tag, and our performance of the *tense* feature is also not comparable. However, for our rest of the experiments TimeBank and TempEval-1 corpus are same, so we will loosely refer to TempEval-1 as TimeBank when comparing with other systems.

The TempEval-1 corpus is divided into a training set of 163 documents and a test set of 20 documents. Since our technique needs far less training data than pure machine-learning based techniques, we used TempEval test data as our development and report the average of 10-fold cross validation performance on the training data, which is totally unseen in our development.

Table 3. Event extraction performance on TempEval with 10-fold cross validation (average)

	Precision	Recall	Fscore	(P+R)/2
TRIPS avg	0.5863	**0.8422**	0.6914	0.7143
TRIOS avg	**0.8327**	0.7168	**0.7704**	0.7748
IAA[a]	N/A	N/A	N/A	0.78

[a] Inter-annotator agreement (IAA) on subset of 10 documents from TimeBank 1.2; they measured the agreement on tag extents by taking the average of precision and recall, which were computed with one annotator's data as the key and the other's as the response. TempEval annotation for EVENT and TIMEX3 were taken verbatim from TimeBank 1.2 (Verhagen et al 2007).

IAA source: http://www.timeml.org/site/timebank/documentation-1.2.html#iaa

Table 3 shows our performance on event extraction. The TRIPS system includes just the TRIPS parser and hand-coded extraction rules. We can see that TRIPS system gets a very high recall but with the expense of precision. Adding the previously discussed MLN filtering step improves the precision by removing the generic and wrong extractions, resulting in the TRIOS system. We get a significant improvement in precision at some cost of some recall. However, overall we have a gain of around 8% in F-score. TimeBank annotators reported their average of Precision and Recall is 78% on event extraction (on a subset of 10 documents). Our TRIOS system's performance is similar to the inter-annotator agreement of the event extraction.

As mentioned already, the TRIPS parser is domain independent and uses extraction rules matching our ontology, which has mapping to WordNet. Hence, porting to a new domain should give similar performance. To see how well TRIPS system does in a new domain, we did an evaluation on two medical text documents (patient reports) with 146 events (human evaluated according to TimeML guideline [4] and found that TRIPS system performed similarly in new domain as well. Our comparison is shown in Table 4.

This performance is suggestive that TRIPS system will have equivalent performance in new domains, but not conclusive because of the small size of the test data. On the other hand, the TRIOS system is dependent on machine learning classifiers, which depends on having a training corpus. So, we cannot get equivalent performance of TRIOS system in new domains without labeled training corpus.

Table 4. Performance of TRIPS system in new (medical) domain vs TRIPS System in old (news) domain

	Precision	Recall	Fscore
TRIPS in TimeBank	0.5863	0.8422	0.6914
TRIPS in Medical Text	0.60	0.83	0.70

Bethard and Martin [6] (the STEP system) had the prior state-of-the-art performance on event extraction in TimeBank corpus. They evaluated their system in 18 documents from the TimeBank corpus and compared with other baselines. The EVITA system by Sauri et al. [5] also performed event extraction system on TimeBank corpus. However, their performance is inflated due to the fact that some aspects of their system were trained and tested on the same data. To get an idea of how well EVITA performs in an unseen data, Bethard and Martin simulated the EVITA system, which they called Sim-Evita. Another of their baseline is Memorize, which assigns the each word the label with which it occurred most frequently in the training data. To compare the performance of our systems, we tested on the STEP test set with the results shown in Table 5.

Table 5. Event extraction performance on Bethard and Martin (2006) paper's test data

	Precision	Recall	Fscore	(P+R)/2
TRIOS	**0.8638**	**0.7074**	**0.7778**	**0.7856**
TRIPS	0.5801	**0.8513**	0.6900	0.7157
STEP [a]	0.82	0.706	0.7587	0.763
Sim-Evita [a]	0.812	0.657	0.727	0.7345
Memorize [a]	0.806	0.557	0.658	0.6815
IAA	N/A	N/A	N/A	0.78

[a] Performances of these systems are reported from Bethard and Martin (2006) paper.

Again the TRIPS system has the highest Recall, which is significantly higher than any other existing systems. But the TRIOS system outperforms all other systems in Fscore. A 10-fold cross validation performance comparison for all systems would have given a better evaluation, but information is not available for the other systems.

We also participated in TempEval-2 challenge. The TRIPS system has the highest recall in TempEval-2 too, and TRIOS system was one of the top systems. Our performance is reported in Table 6 below.

Table 6. Performance of event extraction (Task B) in TempEval-2

	Precision	Recall	Fscore
TRIOS	0.80	0.74	0.77
TRIPS	0.55	0.88	0.68
Best (TIPSem)	0.81	0.86	0.84

7.2. *Event Feature Extraction*

Next, we discuss our performance on event feature extraction. Bethard and Martin [6] report performance for the class of the event. However, they report identifying the *class* and *event* together in terms of precision and recall. This gives an idea of how accurately these features are extracted (precision) and from raw text how many events are extracted with correct *class* feature (recall). We compare these results directly with the TRIOS results in Table 7.

Table 7. Event and class identification performance on Bethard and Martin (2006)'s test set

System	Precision	Recall	Fscore
STEP	0.667	0.512	0.579
TRIOS	**0.780**	**0.551**	**0.650**

Our main gain over the STEP system is in precision, and we also do better than them in recall. In addition to the general linguistically motivated features, our extracted *pos, tense*, *aspect* and suggestions from the TRIPS system are used for identifying the *class,* which improves our performance.

There are also two other systems that report the performance of *class* identification on TimeBank. They are EVITA [5] and Chambers et al. [7], but they evaluate the accuracy ratio, i.e. the percentage of values their system marked correct according to the gold standard. For identifying *class,* EVITA assigns the class for the event that was most frequently assigned to them in the TimeBank. As before, this evaluation is trained and tested on the same document. With this technique they got an accuracy of 86.26%. Chambers et al. [7] also had their majority class baseline, which is same as EVITA, except it doesn't train and test on the same document. Their baseline performance is 54.21%, a better estimate of EVITA's performance on *class* identification.

The remaining three features that we extract are *pos, tense* and *aspect.* As mentioned in the beginning of this section, our experiments are on TempEval corpus, which has different tense values than TimeBank, our performance on *tense* is not directly comparable. TimeBank also didn't have *pos* feature. However, the performance of *aspect* can be compared with other systems. We will be still reporting *tense* performance of other systems and inter-annotator agreement in all cases. Along with these accuracy (precision) numbers, we will also report the recall, which means what percentage of instances we extracted the event and got these features right, i.e. it is strictly dependent on event extraction's accuracy and always lower than that. Our output is gathered from a 10-fold cross validation on the TempEval training data.

EVITA outperforms us, with very small margin, in identification of *aspect* and *tense*, using 140 hand-built lexical rules. But it is important to recall that we are identifying both *nf_morph* and *tense* in the *tense* feature. EVITA's performance on *nf_morph* identification is 89.95%. This means, both systems perform almost equally well in this task. In *pos*, our performance is dependant on third-party pos-tagger software. However,

a naïve baseline method that generates the TimeBank *pos* tags from tagger output has an accuracy of around 87%. Finally, in identifying *class,* we do significantly better than any other existing system.

Table 8. Accuracy or precision of event features and recall of event and event feature extraction (TRIOS on 10 cv on TempEval training data)

	Precision or Accuracy				Recall
Feature	TRIOS 10 cv	C&J 07 10 cv	EVITA	IAA	TRIOS 10 cv
Class	**0.8025**	0.752	0.5421[a]	0.77	0.5749
Tense	0.9105	0.8828[b]	0.9205[b]	0.93	0.6523
Aspect	0.9732	0.9424	**0.9787**	1	0.6973
Pos	**0.9412**	N/A	N/A	0.99	0.6743

[a] The majority class performance is 54.21% for unknown data, from Chambers et al 2007.

[b] Not directly comparable because their corpus had different values for Tense

On the TempEval-2 event feature extraction, our TRIOS system has the best performance on *aspect* and *polarity*. We also do very well (second-best mostly) on *tense*, *class*, *pos* and *modality*. The performance of our systems for event feature extraction in TempEval-2, along with the best performance, is reported in Table 9.

Table 9. Performance of event features (Task B) on TempEval-2

Feature\System	TRIPS	TRIOS	Best in TempEval-2
Class	0.67	0.77	0.79 (TIPSem)
Tense	0.67	0.91	0.92 (Edinburgh-LTG)
Aspect	0.97	0.98	0.98
Pos	0.88	0.96	0.97 (TIPSem, Edinburgh-LTG)
Modality	0.95	0.95	0.99 (Edinburgh-LTG)

7.3. *Identifying Main Event*

We evaluate the performance on main event identification by doing a 10-fold cross validation on TempEval-2 data. We didn't experiment on TimeBank-1, since there's no other system showing performance of main event identification on TimeBank-1. We have several baselines and we explain these in the Table 10. The features mentioned below are already described in the paper.

Table 10. Baseline for main event identification

Name	Features/Description
1. First event baseline (FEB)	Consider the first event of the sentence as the main event
2. First verbal event baseline (FVB)	Consider the first verbal event of the sentence as the main event
3. Hybrid baseline (HYB)	Consider the first verbal event as the main event; if the verbal event doesn't exist in the sentence then consider the first event as the main event
4. Lexical and Penn tag features	Run the classifier with lexical features and penn tag, previous pos and next

without verb word sequence related features (LPV)	pos
5. Lexical and Penn POS tag features (LPF)	Run the classifier with features from 4-LPV + verb word sequence, verb pos sequence, previous word verb sequence, previous pos verb sequence
6. TimeML features generated by TRIOS (TFT)	Run the classifier with TimeML features generated by TRIOS: word, pos, class, tense, polarity, aspect and modality
7. TimeML features taken from corpus (TFC)	Run the classifier with TimeML gold standard features: word, pos, class, tense, polarity, aspect and modality
8. All features (ALF)	Run the classifier with all features – lexical, syntactic and semantic features (described in section 5).

At first we report the performance of all these baselines in Table 11 (first three rows). The classifier based main event extractor doesn't force the constraint that each sentence should have a main event. Hence, after classification, we run a back-off model with our hybrid baseline, i.e. if our classifier doesn't find a main event for a sentence, then it considers the first verbal event as main event and if there are no verbal events then considers the first event as main event. We report the performance of our baselines with back-off hybrid model in last three rows of Table 11. All the features used in the classifiers are generated by our systems, except 7-TFC, where we used the corpus gold standard features.

Table 11. Performance of main event identification and comparison between baselines

	1-FEB	2-FVB	3-HYB	4-LPV	5-LPF	6-TFT	7-TFC	8-ALF
Precision	0.6169	0.6444	0.6485	0.6940	0.7421	0.7120	0.7020	0.7599
Recall	0.4708	0.5423	0.5520	0.6516	0.7013	0.5983	0.5983	0.7299
Fscore	0.5340	0.5890	0.5964	0.6721	0.7211	0.6502	0.6460	**0.7446**
With HYB Backoff								
Precision	X	X	X	0.6802	0.7164	0.6877	0.6807	0.7315
Recall	X	X	X	0.7900	0.8209	0.7534	0.7540	0.8340
Fscore	X	X	X	0.7310	0.7651	0.7190	0.7155	**0.7794**

We observed the following from the experimental results:

1. Our classifier based system with all features (lexical, syntactic and semantic) performed best, which is 15-20% improvement over naïve baselines (1-FEB, 2-FVB and 3-HYB).
2. Incorporating hybrid baseline (3-HYB) as back-off model improved performance (3-6%) for all systems in Fscore.
3. Lexical features and pos tag related features (5-LPF) are good enough for a high performing main event extractor.
   3.1. 5-LPF performs 1-2% less than best performing system with semantic feature (9-ALF); however, if someone wants to extract main events without semantic computation then 5-LPF is a very good option.

3.2. 5-LPF performs better than the system with TimeML featuers (*class, tense, pos, aspect, modality, polarity*), whether system generated (6-TFT) or from the gold standard (7-TFC).
3.3. While just using the Penn tag derived features performs better than using the TimeML features, the verb word sequence related features made the difference. Comparison of 5-LPF and 4-LPV shows that.

7.4. *Recognizing Temporal Expression*

For evaluating our performance on extracting temporal expressions, we again used 10-fold cross validation on the TempEval corpus' training data, our test set. Boguraev and Ando (BA-2005) [13] and Kolomiyets and Moens (KM-2009) [14] also report their performance on TimeBank. Tables 12 and 13 show the comparison between existing systems and our two systems.

Table 12. Temporal expression relaxed match[a] extraction on TimeBank (BA-2005 uses sloppy span[b])

	Precision	Recall	Fscore
KM-2009	0.872	0.836	0.852
BA-2005	0.852	0.952	0.896
CRF+TRIPS	0.8979	0.8951	0.8951
CRF	0.9541	0.8654	**0.9075**

[a] Relaxed match (partial match) admits recognition as long as there are any common words.
[b] Sloppy span admits recognition as long as right boundary is same in the corresponding TimeBank instance.

Table 13. Temporal expression extraction strict match[c] performance on TimeBank

	Precision	Recall	Fscore
KM-2009	0.866	0.796	**0.828**
BA-2005	0.776	0.861	0.817
CRF+TRIPS	0.8064	0.8038	0.8051
CRF	0.8649	0.7846	***0.8228***

[c] Strict match admits recognition when both strings are strictly matched.

We can see that in the relaxed match we outperform existing systems and in strict match we do almost as well as the best state-of-the-art system. However, the performance of our CRF+TRIPS system did not outperform CRF-alone system. To investigate, we hand-checked the extra suggestions of the TRIPS-based system on TempEval test set. We found that among these extra suggestions by TRIPS+CRF system, there are legitimate temporal expressions according to TimeML specification that were missed by the TempEval annotators. In a different work [24], we suggest adding these extra temporal expressions to the TempEval corpus. If we include those temporal expressions, then our TRIPS-based system outperforms our CRF-alone systems by 3-4% in Fscore.

On the TempEval-2 data, both TRIPS and TRIOS use the same CRF based approach. The TempEval-2 scorer handles the matching a bit differently. It counts true positive (tp), true negative (tn), false positive (fp), false negative (fn) and then calculates the precision[t] and recall[u]. For example, if the gold annotation contains a temporal expression *Sunday morning*, and the system's response has *morning,* then there will be one true positive and one false negative (the latter because *Sunday* was not recognized as part of the temporal expression). TempEval-2 scorer performance is shown in Table 14, together with the best scoring system in the TempEval-2 evaluation.

Table 14. Performance of temporal expressions extraction (Task A) on TempEval-2

	TRIOS/TRIPS	Best (HeidelTime-1)
Precision	0.85	0.90
Recall	0.85	0.82
Fscore	0.85	0.8581

7.5. *Determining Normalized Value and Type of Temporal Expressions*

Most previous work on temporal expression extraction on the TimeBank corpus (Boguraev and Ando, 2005 [13] and Kolomiyets and Moens, 2009 [14]) has focused on just recognizing temporal expressions. Boguraev and Ando also report their performance on identifying the *type* of expression (e.g., TIME, DATE, DURATION and SET). We will show the comparison with Boguraev and Ando (BA-2005) on identifying *type.*

We considered the temporal expressions that are matched with the relaxed match and for these instances we checked in how many cases we identified the *type* and *value* accurately. Our 10-fold cross validation performance for both of our systems and performance of BA-2005 on TimeBank is reported in Table 15, which shows we outperform Boguraev and Ando [13].

Table 15. Performance of type and value identification on TempEval for recognized (relaxed) temporal expressions

	type accuracy	*value* accuracy
CRF+TRIPS	0.906	0.7576
CRF	0.9037	0.7594
BA-2005	0.815	N/A

In TempEval-2, our system performs with second best performance on normalization task (identifying *type* and *value*). It is worth mentioning that the average of identifying value performance is 0.61 and if we remove our systems and the best system, HeidelTime-1, the average is only 0.56. Hence, our freely distributed normalization tool

[t] Precision = tp / (tp + fp)
[u] Recall = tp / (tp + fn)

could be beneficiary to many people. Performance of our system and the best system on task A is reported in Table 16.

Table 16. Performance of normalization (*type* and *value* identification – Task A) on TempEval-2

	type accuracy	*value* accuracy
TRIPS/TRIOS	0.94	0.76
Best (HeidelTime-1)	0.96	0.85

8. Conclusion and Future Work

We have presented our work on extracting temporal information from raw text. Our system uses a combination of deep semantic parsing with hand-coded extraction rules, Markov Logic Network classifiers and Conditional Random Filed classifiers. We compared our system with existing systems doing the same task on TimeBank/TempEval corpora. Our system outperforms or does comparably with existing systems. Most importantly, our system performs all these tasks simultaneously within a single unifying framework. In contrast, most of the systems we compared to were trained specifically to do just a single task. However, the performance of our system is best understood in the performance of temporal relation identification. In TempEval-2, our system [26] outperforms other systems in two temporal relation tasks and did equally well in other two, by using these system generated features, whereas other systems used gold corpus features.

We plan to generate larger temporally annotated corpus for the benefit of the research community. We also want to expand our work to other domains. Having a domain independent TRIPS ontology as core for the TRIPS parser, we have showed that for event extraction we have equivalent performance in medical text domains. Ultimately, we also want to build a generic temporally aware system, with performance equivalent to TRIOS system in any domain.

As for the longer term, the event and temporal expression extraction tool is just the first step in building a rich temporal structure of documents. The parser already extracts explicit event-event relationships. We need to evaluate the accuracy of this information, and then build a system that takes the events, event features including tense and aspect, the event relations and explicit temporal expressions, and builds accurate temporal structure of document content.

References

[1] J. Pustejovsky*, et al.*, "The TIMEBANK corpus," 2003.

[2] M. Verhagen*, et al.*, "SemEval-2007 Task 15: TempEval Temporal Relation Identification," presented at the 4th International Workshop on Semantic Evaluations (SemEval 2007), 2007.

[3] J. Pustejovsky and M. Verhagen, "SemEval-2010 task 13: evaluating events, time expressions, and temporal relations (TempEval-2)," presented at the

Workshop on Semantic Evaluations: Recent Achievements and Future Directions, 2010.

[4] J. Pustejovsky, *et al.*, "TimeML: Robust Specication of Event and Temporal Expressions in Text. ," in *New Directions in Question Answering*, 2003.

[5] R. Sauri, *et al.*, "Evita: a robust event recognizer for QA systems," presented at the Human Language Technology and Empirical Methods in Natural Language Processing, 2005.

[6] S. Bethard and J. H. Martin, "Identification of event mentions and their semantic class," presented at the Empirical Methods in Natural Language Processing (EMNLP), 2006.

[7] N. Chambers, *et al.*, "Classifying temporal relations between events," presented at the Association of Computational Linguistics, 2007.

[8] H. Llorens, *et al.*, "TIPSem (English and Spanish): Evaluating CRFs and Semantic Roles in TempEval-2," presented at the International Workshop on Semantic Evaluations (SemEval-2010), Association for Computational Linguistics (ACL), 2010.

[9] C. Grover, *et al.*, "Edinburgh-LTG: TempEval-2 System Description," presented at the International Workshop on Semantic Evaluations (SemEval-2010), Association for Computational Linguistics (ACL), 2010.

[10] D. Ahn, "The stages of event extraction," presented at the Workshop on Annotating and Reasoning about Time and Events, 2006.

[11] C. Aone and M. Ramos-Santacruz, "REES: a large-scale relation and event extraction system," presented at the Sixth conference on Applied natural language processing, 2000.

[12] J.-D. Kim, *et al.*, "Overview of BioNLP'09 Shared Task on Event Extraction," presented at the Proceedings of the Workshop on BioNLP: Shared Task, 2009.

[13] B. Boguraev and R. K. Ando, "TimeBank-Driven TimeML analysis," presented at the Annotating, Extracting and Reasoning about Time and Events, Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005.

[14] O. Kolomiyets and M.-F. Moens, "Meeting TempEval-2: Shallow Approach for Temporal Tagger," presented at the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions, 2009.

[15] D. Ahn, *et al.*, "Extracting Temporal Information from Open Domain Text: A Comparative Exploration," *Digital Information Management,* 2005.

[16] K. Hachioglu, *et al.*, "Automatic Time Expression Labeling for English and Chinese Text," presented at the CICLing, 2005.

[17] J. Poveda, *et al.*, "A Comparison of Statistical and Rule-Induction Learners for Automatic Tagging of Time Expressions in English," presented at the International Symposium on Temporal Representation and Reasoning, 2007.

[18] J. Strotgen and M. Gertz, "HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions," presented at the International Workshop on Semantic Evaluations (SemEval-2010), Association for Computational Linguistics (ACL), 2010.

[19] J. F. Allen, *et al.*, "Deep semantic analysis of text," presented at the Symposium on Semantics in Systems for Text Processing (STEP), 2008.

[20] M. Dzikovska*, et al.*, "Integrating linguistic and domain knowledge for spoken dialogue systems in multiple domains.," presented at the Workshop on Knowledge and Reasoning in Practical Dialogue Systems, IJCAI, 2003.

[21] C. Johnson and C. Fillmore, "The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure," presented at the ANLP-NAACL, Seattle, WA, 2000.

[22] M. Richardson and P. Domingos, "Markov logic networks.," *Machine Learning,* 2006.

[23] S. Riedel, "Improving the accuracy and efficiency of map inference for markov logic," presented at the UAI, 2008.

[24] N. UzZaman and J. F. Allen, "TRIOS-TimeBank Corpus: Extended TimeBank corpus with help of deep understanding of text.," presented at the The seventh international conference on Language Resources and Evaluation (LREC), Malta, 2010.

[25] N. UzZaman*, et al.*, "Multimodal Summarization of Complex Sentence.," presented at the International Conference on Intelligent User Interfaces (IUI), Palo Alto, CA, 2011.

[26] N. UzZaman and J. F. Allen, "TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text," presented at the International Workshop on Semantic Evaluations (SemEval-2010), Association for Computational Linguistics (ACL), Association for Computational Linguistics 2010.