# TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text

**Naushad UzZaman**
University of Rochester
Rochester, NY, USA
naushad@cs.rochester.edu

**James F. Allen**
University of Rochester
Rochester, NY, USA
james@cs.rochester.edu

## Abstract

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. In this paper, we describe two systems we submitted to the TempEval 2 challenge, for extracting temporal information from raw text. The systems use a combination of deep semantic parsing, Markov Logic Networks and Conditional Random Field classifiers. Our two submitted systems, TRIPS and TRIOS, approached all tasks and outperformed all teams in two tasks. Furthermore, TRIOS mostly had second-best performances in other tasks. TRIOS also outperformed the other teams that attempted all the tasks. Our system is notable in that for tasks C – F, they operated on raw text while all other systems used tagged events and temporal expressions in the corpus as input.

## 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 President of Bangladesh in the January of 1983, and we only had documents that tell us about the President from 1980 to 1985 then a temporally aware system will help the QA system to infer who was president in the January of 1983 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 dyslexic people, or people who are not native English speakers, a temporal structure of document could help them to follow the story better. Extracting temporal information will benefit in almost any application processing natural language text.

In this paper, we present the first step towards our goal of building such temporal structures. We participated in all tasks in TempEval 2, which includes work on extracting events, event features, temporal expressions, and various temporal relations.

We first describe our systems. Next, we show the performance of our system and compare with best performing systems on TempEval-2. Finally, we describe our future directions.

## 2 Our System Modules

Our approach for all the tasks is best described as hybrid between linguistically motivated solutions and machine learning classifiers. We do deep semantic parsing and use hand-coded rules to extract events, features and temporal expressions from the logical forms produced by the parser. In parallel, we filter events, extract event features, temporal expressions, classify temporal relations using machine-learning classifiers. We describe these modules briefly here and in the next sections will describe how these modules are used in solving different subtasks.

### 2.1 TRIPS Parser

We use the existing TRIPS parser (Allen et al, 2008) to produce deep logical forms from text. The system is generic and no grammatical rules or lexical entries were added specifically for this task. 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*,



*Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*, pages 276–283,
Uppsala, Sweden, 15-16 July 2010. 

based on rule and lexical weights and the influences of 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 here uses a wide range of statistically driven preprocessing, including part of speech tagging, constituent bracketing, interpretation of unknown words using WordNet, and named-entity recognition (Allen et al, 2008). All these are generic off-the-shelf resources that extend and help guide the deep parsing process.

The TRIPS LF (logical form) ontology[1] 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 (Dzikovska et al. 2003). As much as possible the semantic types in the LF ontology are compatible with types found in FrameNet (Johnson & Fillmore 2000). 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 linguistically motivated ontology[1].

As an example, the result of parsing the sentence, *He fought in the war*, is expressed as 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 of ontology 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*.

### 2.2 Markov Logic Network (MLN)

One of the statistical relational learning (SRL) frameworks that has recently gained momentum as a platform for global learning and inference in AI is Markov Logic (Richardson and Domingos, 2006). Markov logic is a combination of first order logic and Markov networks. It can be seen as a formalism that extends first-order logic to allow formulae to be violated with some penalty.

For our different classification tasks, we used different classifiers based on MLNs. We used an off-the-shelf MLN classifier *Markov thebeast*[2], using Cutting Plane Inference (Riedel, 2008) 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 corresponds to defining features for other machine learning approaches. The Markov network then learns the weights for these formulas from the training corpus and uses 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 two examples that show the learned weights for the formula from training data.

```
tense(e1, INFINITIVE) & aspect(e1, NONE) =>
class(e1, OCCURRENCE)        weight = 0.3199
tense(e1, PRESPART) & aspect(e1, NONE) =>
class(e1, REPORTING) weight = -0.2681
```

The MLN then uses these weights for reasoning 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.

Finding useful features for MLNs is the same as any other ML algorithm. However, the MLN framework gives the opportunity to capture the relations between 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*

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

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

*word*, 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 concatenating the features. But when the size of relations between features increases it complicates matters and we have to regenerate the whole classifier data, every time we introduce a new relationship.

## 3 Event and Event Feature Extraction (Task B)

Because event extraction from the raw text is a prerequisite to everything else we do, we discuss this capability first.

### 3.1 Event 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 minor incremental improvements in rules afterwards. It is worth mentioning, these rules are very generic and can be used in a new domain 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 the 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 above is:

```
<EVENT eid=V2 word=FIGHT
      pos=VERBAL ont-type=FIGHTING
      class=OCCURRENCE tense=PAST
      voice=ACTIVE aspect=NONE
      polarity=POSITIVE
      nf-morph=NONE>
<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 such as ontology type (ont-type). TimeML tries to capture event information by very coarse-grained event features *class* or *pos*. The ontology type feature captures more fine-grained information about the event, but still coarse-grained than the words. The extraction rules also map our fine-grained types to the coarse-grained TimeML event class. We also extract relations between events (SLINK), whenever one event syntactically dominates the other, so it extracts more than TimeML's SLINKs and another new relation, relation between event and its arguments (RLINK). Details about these new additions can be found in UzZaman and Allen (2010).

This system extracts events from the TempEval-2 corpus with high recall. However, this high performance comes with the expense of precision. The reasons for lower precision include, (i) the fact that generic events are not coded as events in TempEval, (ii) errors in parsing and, (iii) legitimate events found by the parser but missed by TempEval annotators. To remedy this problem, we introduced a MLN based filtering classifier, using the event features extracted from the TRIPS parser. The formulas in MLN for filtering were derived by linguistic intuition and by checking the errors in our development set. We devised around 30 formulas.

There were two goals for this filtering step: (1) Eliminating events that result from errors in the parse, and (2) Removing event-classes, such as generics, that were not coded in TempEval.

The second goal is needed to perform a meaningful evaluation on the TempEval challenge. For our long-term goal of using the

For our long-term goal of using the temporal summary in natural language understanding task, however, we would retain these other events. The resulting system, including parsing, extraction, and post-filtering, is named as **TRIOS system**.

### 2.3 Event Feature Extraction

The TRIPS parser and extraction rules already give us event features along with events, which is reported in the results as the **TRIPS system**. To improve performance, we implemented MLN classifiers **(TRIOS system)** for the *class, tense, aspect* and *pos* features, using the features generated from the TRIPS parser plus lexical and syntactical features generated from the text using the Stanford POS tagger[3]. The TRIOS system reports the *polarity* and *modality* performance of TRIPS system, i.e. we don't have any extra classifiers to classify those features in TRIOS system. The Table 1 gives a summary of features used to classify these event features.

| Event feature | Common features | Extra features |
|---|---|---|
| Pos | Event word, event penn tag, verb pos sequence [4], verb word sequence, previous word of verb sequence, previous pos of verb sequence, next word, next pos | none |
| Tense | | pos, polarity, modality, voice (active or passive) |
| Aspect | | pos, polarity, modality, voice (active or passive), pos+previous-pos, pos+next-pos |
| Class | | TRIPS class suggestion, ont-type, slink_core_rel [5], tense+aspect, pos, stem, contains dollar |

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

## 3 Temporal Expression Extraction (Task A)

### 3.1 Recognizing Temporal Expression

The TRIPS parser extracts temporal expressions the same way as we extract events. The performance of TRIPS parser's temporal extraction doesn't outperform state-of-the-art techniques on the evaluation measures. To improve on this, we also use a traditional machine learning classifier straight from the text. We used a token-by-token classification for temporal expressions represented by B-I-O encoding with a set of lexical and syntactic features, using Conditional Random Field (CRF) classifier[6]. CRF is widely used for labeling and segmenting sequence data. Unlike Hidden Markov Models, CRFs are based on exponential models in which probabilities are computed based on the values of a set of features induced from both the observation and label sequences. They have been used in POS tagging, shallow parsing, named entity recognition and also for temporal expression extraction in TERN dataset [Ahn et al. (2005), Hacioglu et al. (2005) and Poveda et al. (2007)].

We used lexical features like *word, shape, is year, is date of week, is month, is number, is time, is day, is quarter, is punctuation, if belong to word-list like init-list*[7]*, follow-list*[8]*,* etc. We then use CRF++ templates to capture relation between different features to extract the sequence. For example, we will write a template to capture the current word is in *init-list* and the next word is in *follow-list*, this rule will train the model to capture sequences like *this weekend*, *earlier morning*, *several years*, etc.

On the other hand, the TRIPS parser does extract temporal relations between events and temporal expressions, which helps us in the temporal relation identification tasks. So we take the temporal expressions from the CRF based extractor and for the cases where TRIPS parser extracts the temporal expression, we use TRIPS parser id, so that we can relate to relations generated by the parser.

The temporal expressions that are suggested by CRF based system, are passed to a filtering step that tries to extract a normalized *value* and *type* of the temporal expression. If we can find a normalized *value* and *type*, we accept the tempo-

[3] http://nlp.stanford.edu/software/tagger.shtml

[4] One Penn tag derived features is *verb word sequence*, 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.

[5] SLINK captures relation between two events, when one syntactically dominates other. This feature captures the relation-type as feature for core events.

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

[7] *init-list* contains words like: *this, mid, first, almost, last, next, early, recent, earlier, beginning, nearly, few, following, several, around, the, less, than, more, no, of, each, late.*

[8] *follow-list* contains words like: *century, centuries, day, days, era, hour, hours, millisecond, minute, minutes, moment, month, months, night, nights, sec, second, time, week, weeks, year, years, am, pm, weekend, summer, fall, winter, fiscal, morning, evening, afternoon, noon, EST, GMT, PST, CST, ago, half.*

ral expressions. We reported this CRF based system with filtering as both **TRIPS and TRIOS systems**.

### 3.2 Determining The Normalized Value and Type of temporal expression

Temporal expressions are most useful for later processing when a normalized *value* and *type* is determined. 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.*

Then in next step we extract the normalized value of temporal expression, as suggested by TimeML scheme. We take the Document Creation Time (DCT) and then calculate the values for different dates in terms of document creation date, e.g. *last month, Sunday, today*. We will make our type and value extractor and temporal expression extractor modules available[9] for public use.

## 4 Temporal Relation Identification (Task C – F)

We identify temporal relations using a Markov Logic Network classifier, namely *thebeast*, by using linguistically motivated features that we extracted in previous steps. Our work matches closely with the work of Yoshikawa et al. (2009). We only consider the local classifiers, but we use more linguistically motivated features and features generated from text, whereas they used TempEval-1's (Verhagen et al., 2007) annotations as input, along with their derived features. Other participants in TempEval 2 also used features from annotated corpus, making us the only group in TempEval-2 to use own-generated entities (events and temporal expression) and features.

TempEval-2 has four subtasks for identifying temporal relations. The tasks are:

(C) Determine the temporal relation between an event and temporal expression in the same sentence;

(D) Determine the temporal relation between an event and the document creation time (DCT);

(E) Determine the temporal relation between the main events in two adjacent sentences; and

(F) Determine the temporal relation between two events, where one event syntactically dominates the other event.

Both TRIPS and TRIOS use the same MLN classifier with same feature-set for each task. However the difference is, they take events and temporal expressions from respective systems, e.g. in Task C (temporal relation between events and temporal expressions), TRIPS system will classify relations for instances where TRIPS event extractor extracted events (in task B) and TRIPS temporal expression extractor extracted temporal expression (in task A). The recall measure of task C will represent the accuracy of extracting events, temporal expression and identifying temporal relations together. This applies for all C – F tasks and for both TRIOS and TRIPS systems.

Tables 2 and 3 show the features we used for each of these tasks. We used some features that we extracted from TRIPS parser. Related information about these concepts can be found in Uz-Zaman and Allen (2010).

| Features | Task C | Task D |
|---|---|---|
| *Event Class* | YES | YES |
| *Event Tense* | YES | YES |
| *Event Aspect* | YES | YES |
| *Event Polarity* | YES | YES |
| *Event Stem* | YES | YES |
| *Event Word* | YES | YES |
| *Event Constituent*[10] | | YES |
| *Event Ont-type* [11] | | YES |
| *Event LexAspect* [12] *x Tense* | | YES |
| *Event Pos* | YES | YES |
| *Timex Word* | | YES |
| *Timex Type* | YES | YES |
| *Timex Value* | YES | YES |
| *Timex DCT relation* | YES | YES |
| *Event to Argument connective ont-type*[13] | YES | YES |
| *Event's argument's ont-type* | YES | YES |
| *TLINK event-time signal*[14] | YES | |

[9] Available online at: http://www.cs.rochester.edu/u/naushad/temporal

[10] TRIPS parser identifies the event constituent along with event word.

[11] Ontology-type is described in Event Extraction subsection.

[12] LexicalAspect feature is a subset of feature *class* and it classifies the events into *Event, State and Reporting* class.

[13] Ontology type of connective that connects the event and its argument

Table 2: Features used for TempEval-2 Task C and D

| Features | Task E | Task F |
|---|---|---|
| *Event Class* | e1 x e2 | e1 x e2[15] |
| *Event Tense* | e1 x e2 | e1 x e2 |
| *Event Aspect* | e1 x e2 | e1 x e2 |
| *Event Polarity* | e1 x e2 | e1 x e2 |
| *Event Stem* | e1 x e2 | e1 x e2 |
| *Event Word* | YES | YES |
| *Event Constituent* | e1 x e2 | e1 x e2 |
| *Event Ont-type* | e1 x e2 | e1 x e2 |
| *Event LexAspect x Tense* | e1 x e2 | e1 x e2 |
| *Event Pos* | e1 x e2 | e1 x e2 |
| *SLINK event-event relation type*[16] | | e1 x e2 |

Table 3: Features used for TempEval-2 Task E and F

# 5 Results

## 5.1 Event and Event Feature Extraction (Task B)

On event extraction, the TRIPS system has the highest recall, while the TRIOS system is second best in precision with the highest scoring system, TIPSem. But overall they do very well compared to our system on event extraction. Performance of our both systems and the best performing TIPSem system is reported in Table 4.

| | Precision | Recall | Fscore |
|---|---|---|---|
| TRIPS | 0.55 | **0.88** | 0.67 |
| TRIOS | 0.80 | 0.74 | 0.77 |
| Best (TIPSem) | **0.81** | 0.86 | **0.84** |

Table 4: Performance of Event Extraction (Task B)

On event feature extraction, our TRIOS system, which is based on MLN based classifiers, has the best performance on *aspect* and *polarity*; we also do very well (second-best performances mostly) on *tense, class, pos* and *modality*.

| Feature | TRIPS | TRIOS | Best |
|---|---|---|---|
| *Class* | 0.67 | 0.77 | 0.79 (TIPSem) |
| *Tense* | 0.67 | 0.91 | 0.92 (Ed.-LTG) |
| *Aspect* | 0.97 | **0.98** | 0.98 |
| *Pos* | 0.88 | 0.96 | 0.97 (TIPSem, Edinburg-LTG) |
| *Polarity* | **0.99** | **0.99** | 0.98 |
| *Modality* | 0.95 | 0.95 | 0.99 (Ed.-LTG) |

Table 5: Performance of Event Features (Task B)

## 5.2 Temporal Expression Extraction (Task A)

Both the TRIPS and TRIOS systems use the same CRF-based approach for temporal expression extraction. Our system has the second best performance on combined temporal expression extraction and 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 could be beneficiary to many people. Performance of our system and the best system on task A is reported in Table 5.

| | | TRIPS TRIOS | Best (HeidelTime-1) |
|---|---|---|---|
| Timex extraction | Precision | 0.85 | 0.90 |
| | Recall | 0.85 | 0.82 |
| Normalization | *type* | 0.94 | 0.96 |
| | *value* | 0.76 | 0.85 |

Table 5: Performance on Temporal Expression extraction (Task A)

## 5.3 Temporal Relation Identification (Task C – F)

For temporal relation identification (Task C – F), other teams used events, temporal expressions and their features from human-annotated corpus, whereas, we used our extracted entities and their features that we extracted in Task A and B. So our performances represent the capability of identifying these relationships from raw text and it is also harder classification task, since we are starting with imperfect features.

Even though we are using our own generated features, we outperformed other groups in task C (temporal relation between event and temporal expression) and task E (temporal relation between main events of consecutive sentences). We also have second-best/equivalent performance for other two tasks (temporal relation between event and DCT; and temporal relation between events, where one syntactically dominates other).

Table 6 reports our systems' performances with precision (P) and recall (R). For others,

[14] TRIPS parser generated event-time TLINK connective or signal (similar to TimeML)

[15] Task E and F is temporal relations between events. In MLN framework, we can write formula in first-order logic. e1 x e2 instances are cases, where we capture both events together. For example, in case of Tense, it will learn the weights for temporal relations given first event's tense is PRESENT and second event's tense is PAST. Instead of just considering first event is PRESENT and second event is PAST, we are considering first event is PRESENT and second event is PAST together.

[16] The SLINK relation type that connects two events, more at UzZaman and Allen (2010).

since they take annotated events and features, they don't actually have a recall, so their recall is not reported.

Since TRIPS system for these tasks uses events (task B) from TRIPS system, which has higher recall, it will have higher recall in relations as well.

| | TRIPS | | TRIOS | | Best |
|---|---|---|---|---|---|
| Task | P | R | P | R | P |
| Task C | 0.63 | **0.52** | **0.65** | **0.52** | 0.65 |
| Task D | 0.76 | **0.69** | 0.79 | 0.67 | 0.82 (TIPSem) |
| Task E | **0.58** | **0.50** | 0.56 | 0.42 | 0.58 |
| Task F | 0.59 | **0.54** | 0.6 | 0.46 | 0.66 (NCSU-indi) |

Table 6: Performance on identifying temporal relations (Task C – F)

### 5.4 Overall Performance

Many teams chose just to attempt one task between task A and B, or both, or only attempt some of tasks C to F. Only three teams attempted all tasks, our team, TIPSem and JU_CSE_TEMP. For tasks C – F, we used our generated features, whereas all other teams used the features provided in the corpus.

In this section, we will show head-to-head comparison of the performance of these three systems to see which team handles the overall challenge of TempEval-2 better. Table 6 summarizes our analysis.

| Task | Description | Best |
|---|---|---|
| Task A | Temporal expression extraction | TRIOS |
| Task B | Event extraction | TIPSem |
| Task C | Event-Timex relationship | TRIOS |
| Task D | Event-DCT relationship | TIPSem |
| Task E | Main event-event relationship | TRIOS |
| Task F | Subordinate event-event relationship | TRIOS |

Table 6: Head-to-head comparison of TRIOS, TIPSem and JU_CSE_TEMP (teams that approached all tasks) in TempEval-2 challenge

Note that JU_CSE_TEMP didn't perform best in any particular task. However, they do a little better than us in Task D (TRIOS 0.79, JU_CSE_TEMP .80). They also didn't extract temporal expression *type* and *value*.

Both TRIOS and TIPSem teams submitted two systems. For this comparison, we pick the best system of each team then compare between them. On temporal expression extraction, we have very close extraction scores (TRIOS fscore 0.85 and TIPSem fscore 0.855). However on temporal expression attributes, we are far superior to TIP-Sem. So overall in Task A, we claim we did better. TIPSem clearly did better on the event extraction task.

On the other hand, given that task A and task B has many subtasks, if we split them into entity extraction and attribute extraction, then we have four tasks of extraction and four tasks on relation identification. In that case, TIPSem does better than us on event extraction, but on event feature extraction we have a tie; for temporal expression extraction, we have another tie, but we outperform in temporal expression attribute extraction.

## 6 Future Work

Our interest is in constructing a domain-independent system for temporal information extraction. We expect that our system will perform closely to TRIPS system (not the better TRIOS) in new domains, since it uses a domain independent semantic parser and domain independent extraction rules. On the other hand, the TRIOS system is dependent on machine learning classifiers, which depends on having a training corpus. So in those cases, we plan to explore bootstrapping a corpus in the new domain using TRIPS, to obtain performance similar to the TRIOS system.

In parallel to that, we plan to build a system that generates reliable temporal structure of documents that can be used in information extraction and language understanding in general. We are particularly interested in generating the temporal structure of documents in medical applications, and in applications that would help people who have trouble reading and understanding documents. To do that, we plan to represent our events, temporal expressions, and temporal relations in a representation like Timegraph (Miller and Schubert, 1990), which is very easy-to-understand representation for humans and also very scalable and efficient solution for temporal reasoning. This would also open the door for applications that require sophisticated temporal reasoning.

Finally, we plan to use the system to semi-automatically create a larger temporally annotated corpus based on TimeML scheme. The sys-

tem would produce an initial version that could be reviewed by human judges before making it public.

## 7 Conclusion

We have shown that a hybrid system combining domain-independent deep understanding techniques with machine learning can extract significant amounts of temporal information from documents. Our submitted systems, TRIPS and TRIOS for TempEval-2 challenge, approached all tasks and outperformed all teams in two tasks (out of six) and TRIOS mostly had second-best performances in other tasks. TRIOS also outperforms the other teams that approached all tasks, even though for task C - F we operated on features automatically computed from raw text rather than using the tagged events and temporal expressions in the corpus.

**Acknowledgments**
This work was supported in part by the National Science Foundation, grant #0748942, and the Office of Naval Research (N000140510314). We thank Mary Swift and William DeBeaumont for help with the TRIPS parser, Benjamin van Durme for many useful suggestions and Sebestian Reidel for help on using the very useful MLN tool *TheBeast*. We are also very thankful to Marc Verhagen and other organizers and annotators of TempEval-2 challenge for organizing a very successful event and also for being very cooperative. Finally, we are thankful to the SemEval-2 chairs, Katrin Erk and Carlo Strapparava for granting us extra pages to describe our approach to all tasks with two systems.

## References

D. Ahn, S.F. Adafre, M. de Rijke, 2005. Extracting Temporal Information from Open Domain Text: A Comparative Exploration, *Digital Information Management,* 2005.

J. Allen, M. Swift, and W. de Beaumont, 2008. Deep semantic analysis of text. In *Symposium on Semantics in Systems for Text Processing (STEP)*, 2008.

M. Dzikovska, J. Allen and M. Swift. 2003. Integrating linguistic and domain knowledge for spoken dialogue systems in multiple domains. *Workshop on Knowledge and Reasoning in Practical Dialogue Systems, IJCAI*, Acapulco.

K. Hachioglu, Y. Chen and B. Douglas, 2005. Automatic Time Expression Labeling for English and Chinese Text. In *Proceedings of CICLing-2005.*

C. Johnson and C. Fillmore. 2000. The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure. *ANLP-NAACL*, Seattle, WA.

J. Poveda, M. Surdeanu and J. Turmo, 2007. A Comparison of Statistical and Rule-Induction Learners for Automatic Tagging of Time Expressions in English. In *Proceedings of the International Symposium on Temporal Representation and Reasoning, 2007.*

S.A. Miller and L.K. Schubert, Time Revisited, *Computational Intelligence 6*(2), 108-118, 1990.

James Pustejovsky, Jos M. Castao, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev, 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. In Mark T. Maybury, editor, *New Directions in Question Answering*, pages 28–34. AAAI Press, 2003.

Matthew Richardson and Pedro Domingos, 2006. Markov logic networks. *Machine Learning*, 2006.

Sebastian Riedel. 2008. Improving the accuracy and efficiency of map inference for markov logic. *In Proceedings of UAI 2008.*

Naushad UzZaman and James Allen, 2010, TRIOS-TimeBank Corpus: Extended TimeBank corpus with help of Deep Understanding of Text, To Appear in the Proceedings of *The seventh international conference on Language Resources and Evaluation (LREC), Malta, 2010.*

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz and James Pustejovsky, 2007. SemEval-2007 Task 15: TempEval Temporal Relation Identification, Proceedings of *4th International Workshop on Semantic Evaluations (SemEval 2007)*.

Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara and Yuji Matsumoto. 2009. Jointly Identifying Temporal Relations with Markov Logic. Proceedings of the *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2009.*