Extracting Events and Temporal Expressions from Text

Naushad UzZaman
Computer Science Department
University of Rochester
Rochester, NY, USA
naushad@cs.rochester.edu

James F. Allen
Computer Science Department
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. Our long-term goal is to build complete temporal structure of documents and apply the temporal structure in other applications like textual entailment, question answering, dialog systems or others. In this paper, we present a first step, a system for extracting event, event features, temporal expression and its 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 TimeBank corpus. Our system outperforms or does equally well with all existing systems that evaluate on the TimeBank corpus and our performance is very close to inter-annotator agreement of the TimeBank annotators.**

Keywords - event extraction; temporal expression extraction; TimeBank; TempEval; temporal information processing; information extraction; TRIPS; TRIOS

I. 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 dyslexic people 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 information, we start with describing related efforts. Then, we describe our system to extract temporal information. Next, we show the performance of our system and compare with other systems on TimeBank [15] corpus. Finally, we conclude describing our future direction.

II. RELATED WORK

To a first approximation, our view of events matches closely with the TimeML (Pustejovsky et al.) [14] 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.

There have been a few existing systems that implemented and tested their systems based on TimeBank to compare with competitive systems. Sauri et al. [19] 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 [5] 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 and reported their performance on TimeBank.

There has been other work on event extraction on other corpora. One significant effort includes systems like Ahn [1], Aone and Ramos-Santacruz [4] and many others, which are based on or related to MUC-7[1] or ACE[2] specifications, which specifies the task as not just extracting event but also extracting event arguments, assigning roles and determination of event coreference. But in these cases the entities are limited and pre-defined set like person, organization, location, geo-political entity, facility, vehicle, weapon, etc. We are interested in all events like TimeML, instead of limited set of entities with detailed information.

The other line of significant work is event extraction in Biological domains, especially the BioNLP shared task on event extraction [11]. Their event extraction is defined based

[1] Message Understanding Conference
[2] http://www.nist.gov/speech/tests/ace/

on the biological domain dependent GENIA[3] 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. There was a team [18] that used Markov Logic Networks, which we use in many of our sub-problems. Our difference with them is, i) they use MLN for their all tasks, whereas we use it in different sub-problems; ii) they handle the different tasks jointly, e.g. extracting events and its arguments, whereas we extract these arguments with semantic parser and use the MLN to filter out extracted events or to extract event features. We couldn't compare with their system because their task is on different corpus.

On the other hand, most of the state-of-the-art systems that extract temporal expressions were developed in the scope of ACE Temporal Expression Recognition and Normalization[4], in which TIMEX2 tags denote temporal expression. 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 [6] and Kolomiyets and Moens [12] 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. [2], Hacioglu et al. [9] and Poveda et al. [13] used approaches with a token-by-token classification for temporal expressions represented by a B-I-O encoding with lexical and syntactic features and tested on the TERN dataset.

We extract both events and temporal expressions. Hence, we evaluate our system for both events and temporal expression extraction together on temporally annotated TimeBank corpus.

III. Our System Modules

Our approach for event extraction is linguistically motivated; we do deep semantic parsing and extract events and features using hand-build rules that do graph matching on the logical forms. In the next filtering step, we use a Markov Logic Network (MLN) classifier to filter out errors. We also implemented MLN classifiers to directly classify *class, tense, aspect* and *pos* feature from surface features extracted from the text. For temporal expressions, we use both TRIPS parser's extraction and a Conditional Random Field based classifier. Before describing these techniques, we will describe two significant modules of our system, the TRIPS parser and the Markov Logic Network classifier.

A. TRIPS Parser

We use the TRIPS parser (Allen et al) [3] 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[5] 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.) [8]. 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 (Johnson and Fillmore) [10]. 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

[3] http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/topics/Corpus/genia-ontology.html
[4] http://fofoca.mitre.org/tern.html
[5] TRIPS ontology browser: http://www.cs.rochester.edu/research/trips/lexicon/browse-ont-lex.html

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.

B. 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 (Richardson and Domingos) [16]. Markov logic is a combination of first order logic and Markov networks. It can be assumed 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*[6], using Cutting Plane Inference [17] 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 finding 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*, 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.

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

IV. 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
      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, 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 (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 [20].

The TRIPS parser extracts events from the TimeBank corpus with very high recall compared with any other existing system. 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 TimeBank (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 this problem, we introduced a MLN based filtering classifier, using the event features extracted from TRIPS parser. There are two goals for this filtering step:

1) Eliminate events that result from errors in the parse,

2) For evaluation on Timebank, Remove some event-classes, such as generics, that were not coded in Timebank.

As noted, the second goal is needed to perform a meaningful evaluation on the Timebank corpus. For our long-term goal of extracting the temporal structure of documents, however, we would retain these other events. The resulting system, including parsing, extraction, and post-filtering, is named as TRIOS system.

TABLE I. 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	Not available
Tense		pos, polarity, modality, voice (active or passive)
Aspect		pos, polarity, modality, voice (active or passive), pos x previous pos, pos x next pos
Class		TRIPS class suggestion, ont-type, tense x aspect, pos, stem, contains dollar

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 *class, tense, aspect* and *pos* feature, using the features generated from the TRIPS parser plus lexical and syntactic features generated from the text using the Stanford POS tagger. Table I gives a summary of attributes used to classify these event features.

V. TEMPORAL EXPRESSION EXTRACTION

A. 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 B-I-O encoding with a set of lexical and syntactic features, using Conditional Random Field classifier[7]. 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, we have found that TRIPS extracts some temporal expressions that are missed by our CRF based system and even sometimes missed by TimeBank annotators. So we implemented a system by making a hybrid between CRF based system and TRIPS suggestion.

The temporal expressions that are suggested by the TRIPS parser but are missed by 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.

B. 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 is also a subtask for 2010 TempEval challenge[8].

TABLE II. 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 extract the normalized value of temporal expression, as suggested by TimeML scheme. We take the

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

[8] http://www.timeml.org/tempeval2/

Document Creation Time (DCT) from the documents and then calculate the values for different dates, e.g. last month, Sunday, today. Other *type* instances are trivial. Table 4 shows some temporal expressions examples with normalized *value* and *type*.

VI. Evaluation and Discussion

A. Event Extraction

As mentioned before, the TimeBank corpus [15] is annotated according to TimeML specification. Later in TempEval (Temporal Evaluation contest) (Verhagen et al.) [21], they released the same corpus with modified event relations and modifying some 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.

And finally, event nominalizations that don't provide any extra information than the supplied verb are also not tagged.

As for event attributes, TimeML considers *class, tense, aspect, and nf_morph (Non-finite morphology).* They use only seven abstract event classes rather than the detailed classes in the TRIPS ontology:

1) *Occurrence: die, crash, build;*
2) *State: on board, kidnapped;*
3) *Reporting: say, report;*
4) *I-Action: attemt, try, promise;*
5) *I-State: believe, intend, want;*
6) *Aspectual: begin, stop, continue;*
7) *Perception: see, hear, watch, feel.*

For *tense*, they consider *PAST, PRESENT, FUTURE, NONE*, for *aspect*, they consider *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 being 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*.

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

The TempEval corpus is divided into a training set of 163 documents and a test set of 20 documents. Because our technique needs only a little training data, we used TempEval test data as our development set for event extraction, and report the average of 10 cross-fold validation performance on the training data, which is totally unseen in our development.

TABLE III. Event Extraction Performance on TempEval with 10 cross-fold 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[9]	N/A	N/A	N/A	0.78

Table III 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. But, as mentioned already, the TRIPS parser is domain independent and uses extraction rules matching our ontology, which has mapping to WordNet. Hence, porting in 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 [14] and found that TRIPS system performed similarly in new domain as well. Our comparison is shown in Table IV.

This performance is suggestive that TRIPS system will have equivalent performance in new domains, but not conclusive, since it was tested in just two documents with 146 events in medical texts. On the other hand, the better TRIOS system, which is explained next, is dependent on machine learning classifiers, which depends on having a training corpus.

[9] 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

So, we cannot get equivalent performance of TRIOS system in new domains without labeled training corpus.

TABLE IV. 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

To compare with other systems in TimeBank, we implemented the previously discussed filtering step to improve the precision by trying to remove the generic and wrong extractions. This combination of TRIPS system and MLN filtering is the TRIOS system, where 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.

Bethard and Martin [5] (STEP system) had the prior state-of-the-art performance on event extraction in TimeBank corpus. They evaluated their system in 18 documents from TimeBank corpus and compared with other baselines. The EVITA by Sauri et al. [19] also implemented the 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 in the STEP test set with the results reported in Table V.

TABLE V. 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*	0.82	0.706	0.7587	0.763
Sim-Evita*	0.812	0.657	0.727	0.7345
Memorize*	0.806	0.557	0.658	0.6815
IAA	N/A	N/A	N/A	0.78

Again the TRIPS system has the highest Recall, which is ~15% higher than any other existing systems. But the TRIOS system outperforms all other systems in Fscore. STEP's recall is closest and almost similar to us, but we gain in MLN filtering step and end up with an overall higher precision.

A 10-cross validation performance comparison for all systems would have given a better evaluation, but information is not available for the other systems.

* Performances of these systems are reported from Bethard and Martin (2006) [5] paper.

B. *Event Feature Extraction*

Next, we discuss our performance on event feature extraction. Bethard and Martin [5] only report performance for event feature *class*; however, they report identifying *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 VI.

TABLE VI. 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 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 [19] 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 cross validation on the TempEval training data.

EVITA outperforms us, with very small margin, in identification of *aspect* and *tense*, 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 VII. 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[10]	0.77	0.5749
Tense	0.9105	0.8828**	0.9205**	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

C. Recognizing Temporal Expression

For evaluating our performance temporal expressions, we again used 10 cross validation on the TempEval corpus' training data, our test set. Boguraev and Ando (BA-2005) [6] and Kolomiyets and Moens (KM-2009) [12] also report their performance on TimeBank. Tables VIII and IX show the comparison between existing systems and our two systems.

TABLE VIII. TEMPORAL EXPRESSION RELAXED MATCH[11] EXTRACTION ON TIMEBANK (BA-2005 USES SLOPPY SPAN[12])

	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**

TABLE IX. TEMPORAL EXPRESSION EXTRACTION STRICT MATCH[13] 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***

We can see that in the relaxed match we outperform existing systems and in strict match we do almost as well with the best state-of-the-art system. However, our CRF with TRIPS system's performance 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 paper [20], we suggest adding these extra temporal expressions to the TempEval corpus. However, for this work, if we include those temporal expressions, then our TRIPS-based system outperforms our CRF-alone systems by 3-4% Fscore. On the other hand, CRF+TRIPS based system also has an edge over CRF-only system in recall, which means this system will be preferable for automatic temporal annotation with human review, where human reviewers can easily remove the errors.

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

** Not directly comparable because their corpus had different values for Tense

[11] Relaxed match admits recognition as long as there are any common words.

[12] Sloppy span admits recognition as long as right boundary is same in the corresponding TimeBank instance.

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

D. Determining Normalized Value and Type of Temporal Expressions

Most previous work on temporal expression extraction on the TimeBank corpus (Boguraev and Ando, 2005 [6] and Kolomiyets and Moens, 2009 [12]) have focused on just recognizing temporal expressions. Boguraev and Ando also report their performance on identifying *type*. We will show the comparison with Boguraev and Ando (BA-2005) on identifying *type*. No other system to date[14] reports results on computing normalized *values.*

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 cross-fold validation performance for both of our systems and performance of BA-2005 on TimeBank is reported in Table X, which shows we outperform Boguraev and Ando [6].

TABLE X. 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

VII. CONCLUSION AND FUTURE WORK

We presented our work on extracting temporal information from raw text. Our system is 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 corpus. Our system outperforms existing systems in event extraction, temporal expression relaxed match, temporal expression type and normalized value identification and event *class* identification and does equally well as existing systems in other event features and exact-match temporal expressions.

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

As for our long-term future work, the event and temporal expression extraction tool is just the first step in building a rich temporal structure of documents. The parser is already extracting 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.

[14] TempEval-2010 has a task on identifying these normalized values, which is not published yet.

REFERENCES

[1] David Ahn. The stages of event extraction, *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, July 2006. Association for Computational Linguistics.

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

[3] James Allen, Mary Swift, and Will de Beaumont. Deep semantic analysis of text. In *Symposium on Semantics in Systems for Text Processing (STEP)*, 2008.

[4] Chinatsu Aone and Mila Ramos-Santacruz. REES: a large-scale relation and event extraction system, *Proceedings of the sixth conference on Applied natural language processing*, 2000.

[5] Steven Bethard and James H. Martin. Identification of event mentions and their semantic class. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

[6] B. Boguraev and R. K. Ando. TimeBank-Driven TimeML analysis. In *Annotating, Extracting and Reasoning about Time and Events*, Dagstuhl Seminar Proceedings. Dagstuhl, Germany, 2005.

[7] N. Chambers, S.Wang, and D. Jurafsky. Classifying temporal relations between events. Proceedings of the *ACL 2007*.

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

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

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

[11] Jin-Dong Kim, Tomoto Ohta, Sampo Pyysalo, Yoshinabu Kano, Jun'ichi Tshujii. Overview of BioNLP'09 Shared Task on Event Extraction, *Proceedings of the Workshop on BioNLP: Shared Task*, pages 1–9, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[12] O. Kolomiyets and Marie-Francine Moens. Meeting TempEval-2: Shallow Approach for Temporal Tagger, *In Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions, 2009.*

[13] J. Poveda, M. Surdeanu and J. Turmo. 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*.

[14] James Pustejovsky, Jos M. Castao, Robert Ingria, Roser Sauri, Robert J. Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R. Radev. 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.

[15] J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*, pages 647–656, March 2003.

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

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

[18] Sebastian Riedel, Hong-Woo Chun, Roshihisa Takagi and Jun'ichi Tsujii. A Markov Logic Approach to Bio-Molecular Event Extraction, *Proceedings of the Workshop on BioNLP: Shared Task, pages 41–49,* Boulder, Colorado, June 2009. Association for Computational Linguistics.

[19] Roser Sauri, Robert Knippen, Marc Verhagen, and James Pustejovsky. Evita: a robust event recognizer for QA systems. *In HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 700–707, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[20] 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.

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