# A Supervised Machine Learning Approach for Temporal Information Extraction

Anup Kumar Kolya[a], Asif Ekbal[b], and Sivaji Bandyopadhyay[c]

[a, c] Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India
anup.kolya@gmail.com[a], sivaji_cse_ju@yahoo.com[c]

[b] Department of Information Engineering and Computer Science, University of Trento, Italy
asif.ekbal@gmail.com[b]

**Abstract.** Temporal information extraction is an interesting research area in Natural Language Processing (NLP). Here, the main task involves identification of the different relations between various events and time expressions in a document. The relations are then classified into some predefined categories like BEFORE, AFTER, OVERLAP, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER and VAGUE. In this paper, we report our works of temporal information extraction along the lines of TempEval-2007 evaluation challenge. We adapt supervised machine learning approach for solving the problems of all the three tasks, namely A, B and C. Initially, a *baseline* system is developed by considering the most frequent temporal relation in the corresponding task's training data. Evaluation results on the TempEval-2007 datasets yield the F-score values of 59.8%, 73.8% and 43.8% for Tasks A, B and C, respectively under the strict evaluation scheme. All these systems show the F-score values of 61.1%, 74.8% and 46.9% for Tasks A, B and C, respectively under the relaxed evaluation scheme. For the *sub-ordinate* event in Task C, the system shows the F-score values of 55.1% and 56.9% under the strict and relaxed evaluation scheme, respectively.

**Keywords:** Temporal Relation Identification, TimeML, Conditional Random Field, TempEval-2007, Tasks A, B and C.

## 1 Introduction

Temporal information extraction has become a popular and interesting research area of Natural Language Processing (NLP) since the last few years. Generally, events are described in different newspaper texts, stories and other important documents where events happen in time. Many text processing applications require identifying these events described in a text and locating these in time. This is also important in a wide range of NLP applications that include temporal question answering, machine translation and document summarization etc. The TempEval-2007 challenge (Verhagen *et al.*, 2007) addressed this question by establishing a common corpus on which various research teams came up with different approaches to find temporal relations. In TempEval-2007, the following types of temporal relations (i.e. event-event and/or event-time) were considered:

- **Task A:** Relation between the events and times within the same sentence.
- **Task B:** Relation between events and document creation times.
- **Task C.** Relation between the verb events in adjacent sentences.

In each of these tasks, systems attempted to annotate appropriate pairs with one of the relations, namely BEFORE, BEFORE-OR-OVERLAP, OVERLAP, OVERLAP-OR-AFTER, AFTER or VAGUE. The participating teams were instructed to find all temporal relations of these types in a corpus of newswire documents.

In the literature, temporal relation identification has been treated as a classification problem and solved using machine learning in a number of proposals (Boguraev *et al.*, 2005; Mani *et al.*, 2007; Mani *et al*., 2007; Chambers *et al.*, 2007). Some of the TempEval-2007 participants

(Verhagen *et al*., 2007) also proposed several machine learning based approaches to identify and classify different temporal relations. In TempEval-2007 (Verhagen *et al.*, 2007) task, a common standard dataset was introduced that involves three temporal relations. The participants reported F-scores values for event-event relations ranging from 42% to 55% and for event-time relations from 73% to 80%.

In our present work, we propose supervised machine learning approaches to solve the problems of all the three tasks, i.e. A, B and C of TempEval-2007. The task of temporal relation identification is considered as a pair-wise classification problem, where each event/time, event/document creation time or event/event pair is assigned one of the TempEval relation classes (i.e. BEFORE, AFTER, etc.). Event/time pairs are encoded using syntactically and semantically motivated features in the TimeBank corpus. These features are automatically extracted from the training data and used to train a supervised machine learning model, Conditional Random Field (CRF). It is to be noted that we only used the features available in the training/test datasets.

The remainder of this paper is structured as follows. Section 2 describes about the tasks. Section 3 presents our proposed approach that discusses very about CRF in brief, various features, used to characterize the various relationships between event and time expressions, and the various steps of the overall system architecture. Evaluation scheme is presented in Section 4. Detailed experimental results are reported in Section 5. Finally, Section 6 concludes the paper.

## 2 Description of Task

In TempEval-2007, three different tasks were defined to identify various temporal relations from the text and classify them into some predetermined categories. Task A at TempEval 2007 was involved with automatic identification of the temporal relations holding between events and all temporal expressions appearing in the same sentence. The main problem of Task B was to find out the relations between the events and document creation time. Task C was involved with the automatic identification of temporal relations holding between verb events in adjacent sentences. This task addresses only the temporal relations holding between time and event expressions that occur within the same and/or consecutive two sentence(s). In each sentence, only one main event is identified. If any sentence has only one event, then it automatically becomes the main event of that sentence. In the case of sentences with multiple events, the main event is determined following very shallow, syntactic-based criteria.

The events expressions (TEs) were annotated in the source in accordance with the TimeML standard (Pustejovsky *et el.*, 2003). For all the tasks, data were provided for training and testing that includes annotations identifying: (1) sentence boundaries, (2) all temporal referring expression as specified by TIMEX3, (3) all events as specified in TimeML and (4) selected instances of temporal relations, as relevant to the given task. For all the tasks, a restricted set of event terms were identified–those whose stems occurred twenty times or more in TimeBank. This set is referred to as the Event Target List or ETL. Furthermore, only event expressions that occur within the ETL are considered. In the training and test data, TLINK annotations for these temporal relations are provided. The only difference being that in the test data the relation type is withheld. The task is to supply this label.

## 3 Conditional Random Field Based Approach

Our approach for temporal relation identification and classification is based on a supervised machine learning algorithm, namely Conditional Random Field (CRF). This is capable to include arbitrary set of features, but can still avoid overfitting in a principled manner. We consider the task as a pair-wise classification problem in which the target pairs–Event-Time, Event-Document Creation Time and Event-Event are modeled using CRF.

### 3.1 Conditional Random Field

Conditional Random Field (CRF) (Lafferty *et el.*, 2001) is an undirected graphical model, which is a special case of which corresponds to conditionally trained probabilistic finite state automata. Being conditionally trained, these CRFs can easily incorporate a large number of arbitrary, non-independent features while still having efficient procedures for non-greedy finite-state inference and training. The main advantage of CRF comes from that it can relax the assumption of conditional independence of the observed data often used in generative approaches, an assumption that might be too restrictive for a considerable number of object classes. Additionally, CRF avoids the label bias problem.

CRF is used to calculate the conditional probability of values on designated output nodes given values on other designated input nodes. The conditional probability of a state sequence $S =< s_1, s_2, ..., s_T >$ given an observation sequence $O =< o_1, o_2, ....., o_T)$ is calculated as:

$$P_\Lambda(s \mid o) = \frac{1}{Z_o}\exp(\sum_{t=1}^{T}\sum_{k=1}^{K}\lambda_k f_k(s_{t-1}, s_t, o, t))$$

where, $f_k(s_{t-1}, s_t, o, t)$ is a feature function whose weight $\lambda_k$ is to be learned via training. The values of the feature functions may range between $-\propto .....+\propto$, but typically they are binary. To make all conditional probabilities sum up to 1, we must calculate the normalization factor,

$$Z_0 = \sum_s \exp(\sum_{t=1}^{T}\sum_{k=1}^{K}\lambda_k f_k(s_{t-1}, s_t, o, t)),$$

this, as in HMMs, can be obtained efficiently by dynamic programming.

To train a CRF, the objective function to be maximized is the penalized log-likelihood of the state sequences given the observation sequence:

$$L_\wedge = \sum_{i=1}^{N}\log(P_\wedge(s^{(i)} \mid o^{(i)})) - \sum_{k=1}^{K}\frac{\lambda_k^2}{2\sigma^2},$$

where, $\{< o^{(i)}, s^{(i)} >\}$ is the labeled training data. The second sum corresponds to a zero-mean, $\sigma^2$ -variance Gaussian prior over parameters, which facilitates optimization by making the likelihood surface strictly convex. Here, we set parameters $\lambda$ to maximize the penalized log-likelihood using Limited-memory BFGS (Sha and Pereira, 2003), a quasi-Newton method that is significantly more efficient, and results in only minor changes in accuracy due to changes in $\sigma$.

CRFs generally can use real-valued functions but it is often required to incorporate the binary valued features. A feature function $f_k(s_{t-1}, s_t, o, t)$ has a value of 0 for most cases and is only set to 1, when $s_{t-1}, s_t$ are certain states and the observation has certain properties.

We have used the C$^{++}$ based CRF++ package[1], a simple, customizable, and open source implementation of CRF for segmenting /labeling sequential data.

### 3.2 Temporal Features Used for CRF Training and Testing

We use the gold-standard TimeBank features for training and testing the CRF model. These features are extracted automatically from the respective datasets. Here, we mainly use various combinations of the subsets of the following features:

[1]http://crfpp.sourceforge.net

(i).**Event class**: This is denoted by the 'EVENT' tag and used to annotate those elements in a text that mark the semantic events.
(ii). **Event stem**: This denotes the stem of the head event
(iii). **Event and time strings**: This denotes the actual event strings and time.
(iv). **Part of Speech of event terms**: POS information is very useful to identify various temporal relations. The features values may be either of ADJECTIVE, NOUN, VERB, and PREP.
(v). **Event tense**: This feature is useful to capture the standard distinctions among the grammatical categories of verbal phrases. The tense attribute can have values, namely PRESENT, PAST, FUTURE, INFINITIVE, PRESPART, PASTPART or NONE.
(vi). **Event aspect**: This feature denotes the aspect attribute of event that may take values**,** PROGRESSIVE, PERFECTIVE and PERFECTIVE PROGRESSIVE or NONE.
(vii). **Event polarity**: Polarity of an event instance, represented by the boolean values POSITIVE or NEGATIVE.
(viii). **Event modality**: The modality attribute is only present if there is a modal word present that modifies the instance.
(ix) **Type of temporal expression**: It represents the temporal relationship holding between the various event and time expressions.
(x). **Temporal signal**: This feature represents the various temporal prepositions.
(xii). **Temporal expression in the target sentence**: This feature takes the values *greater than*, *less than*, *equal* or *none*.

We use the following subsets of features for each task. All the listed features from (i)-(xii) are for Task A; POS, event tense, event aspect and temporal expression in the target sentence are for Task B; and event class, POS, event tense, event aspect and temporal expression in the target sentence are for Task C.

## 3. 3 Various Steps for CRF based Relation Extraction

We use supervised learning algorithm, CRF, for identifying various temporal relations and classifying them into some predefined categories. The input is a document (i.e., training or test set). We obtain this dataset from the TempEval-2007 evaluation task. The dataset is preprocessed for the specified CRF format. Thereafter, we extract the features in the form of vectors from the annotated training data. A feature vector consisting of the available features as described in Section 3.2 is extracted for each <*event*, *time*> pair in Task A, <*event, document creation time*> pair in Task B and <*event, event*> pair in Task C from the TimeBank corpus. The <*event, event*> pair in Task C could be the pairs of <*main-event*, *main-event*> and <*main-event*, *next subordinate event*, *previous subordinate event*, *main-event*>. Now, we have a training data in the form $(W_i, T_i)$, where, $W_i$ is the $i^{th}$ pair along with its feature vector and $T_i$ is its corresponding TempEval relation class.

All the feature vectors are extracted from the training data. The first attribute denotes <*event*, *time*> pair in Task A, <*event, document creation time*> pair in Task B and <*event, event*> pair in Task C. The last attribute denotes the corresponding temporal relationship type. The temporal relation is annotated by one of the output labels, such as BEFORE, BEFORE-OR-OVERLAP, OVERLAP, OVERLAP-OR-AFTER, AFTER or VAGUE. The remaining attributes of the feature vector denote the features. Then, we train the CRF model using the automatically extracted feature vectors and by defining the appropriate feature template. Feature template defines the probabilities.

Models are created from the training set and the feature template. The same feature extraction methodology is again repeated for the test data. An unknown instance of <*event*, *time*>, <*event, document creation time*> or <*event, event*> is assigned the appropriate output label, i.e. OVERLAP, BEFORE, AFTER, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER and VAGUE, depending upon the probabilities, learned in the CRF model. The output label predicted by the CRF is matched against the reference label.

## 4 Evaluation Scheme

For TempEval -2007, the tasks were defined in such a way that a simple pairwise comparison is possible since it was not required to create a full temporal graph and judgments are made in isolation. The organizers used two scoring schemes: **strict** and **relaxed**.

The strict scoring scheme only counts exact matches as success. For example, if the key is OVERLAP and the response is BEFORE-OR-OVERLAP then this is counted as 'failure'. The standard definitions of precision and recall are followed:

$$\text{Pr}\,ecision = Rc / R$$

$$\text{Re}\,call = Rc / K$$

where, $Rc$ is number of correct answers in the response, $R$ is the total number of answers in the response, and $K$ *is* the total number of answers in the key.
For the relaxed scoring scheme, precision and recall are defined as

$$\text{Pr}\,ecision = Rcw / R$$

$$\text{Re}\,call = Rcw / K$$

where, $Rcw$ reflects the weighted number of correct answers. The *F-score* is measured as follows where Pr = Pr*ecision* and Re = Re*call*:

$$F - score = 2\,\text{Pr}^{*}\,\text{Re}/\left(\text{Pr}+\text{Re}\right)$$

## 5 Experimental Result and Discussions

For each of the tasks, we develop a number of CRF models depending upon the various features included into it. We have a training data in the form $(W_i, T_i)$, where, $W_i$ is the $i^{th}$ pair along with its feature vector and $T_i$ is its corresponding TempEval relation class. Models are built based on the training data and the feature template. The procedure of training is summarized below:

1. Define the training corpus, C.
2. Extract the <*event*, *time*>, <*event*, *document creation time*> and <*event*, *event*> relations from the training corpus.
3. Create a file of candidate features derived from the training corpus.
4. Define a feature template.
5. Compute the CRF weights $\lambda_k$ for every $f_K$ using the CRF toolkit with the training file and feature template as input.
6. Derive the best feature template depending upon the performance.
7. Select the best feature template obtained from Step 6.
8. Retrain the CRF model

We use various subsets of the following feature template during our experiment. In the figure, $w_i$ : Current <*event, time*>, <*event, document creation time*> and <*event, event*> pair, $w_{(i-n)}$ : Previous nth pair, $w_{(i+n)}$ : Next nth pair, $t_{i-1:}$ previous pair.

| |
|---|
| $w_{(i-2)}$ |
| $w_{(i-1)}$ |
| $w_i$ |
| $w_{i+1}$ |
| $w_{(i+2)}$ |
| Combination of $w_{i-1}$ and $w_i$ |
| Combination of $w_i$ and $w_{i+1}$ |
| Dynamic output tag ($t_i$) of the previous pair |
| Feature vector of $w_i$ of other features |

**Figure 1:** Feature template used for the experiment

## 5.1 Results of Task A

The following table, Table 1 illustrates the results for Task A. The system demonstrates the precision, recall and F-score of 59.8%, 59.8% and 59.8% under the strict evaluation scheme. Under relaxed evaluation, the system yields precision, recall and F-score of 61.1%, 61.1% and 61.1%, respectively. The *baseline* is based on the most frequent temporal relation encountered in the training data of the task. In the case of task A, the most frequent temporal relation present in the training data is OVERLAP. The CRF based system performs better than the *baseline* model with more than 3% F-score under the strict evaluation framework and 4.03% F-score under the relaxed evaluation framework.

**Table 1: Evaluation results for Task A (we report percentages)**

| Model | Evaluation scheme | Precision | Recall | F-score |
|---|---|---|---|---|
| Baseline | Strict | 56.8 | 56.8 | 56.8 |
| | Relaxed | 58.9 | 58.9 | 58.9 |
| CRF | Strict | 59.8 | 59.8 | 59.8 |
| | Relaxed | 61.1 | 61.1 | 61.1 |

## 5.2 Results of Task B

Various experiments are carried out with the different representations of the feature template as shown in Figure 1. Results show that the system performs best with the context of size five (i.e., previous two, current and the next two <event, DCT> pairs), tense and aspect features. It shows the precision, recall and F-score values of 71.4%, 71.0% and 71.2%, respectively under the strict evaluation scheme and 71.8%, 71.3% and 71.5%, respectively under the relaxed evaluation scheme. The overall evaluation results of the system are presented in Table 2. The *baseline* model is developed based on the most frequent temporal relation encountered in the training data for the task. In the case of task B, the most frequent temporal relation present in the training data is BEFORE. Results show that the CRF based system performs better than the

*baseline* model with the margins of 16.7% F-score in the strict evaluation scheme and 16.9% F-score in the relaxed evaluation scheme.

**Table 2: Evaluation results for Task B (we report percentages)**

| Model | Evaluation scheme | Precision | Recall | F-score |
|---|---|---|---|---|
| Baseline | Strict | 57.1 | 57.1 | 57.1 |
| | Relaxed | 57.9 | 57.9 | 57.9 |
| CRF | Strict | 74.1 | 73.6 | 73.8 |
| | Relaxed | 75.1 | 74.6 | 74.8 |

## 5.3 Results of Task C

Initially, we experiment with the several feature templates (of Figure 1) for identifying the various temporal relations related either to *main-event* or *sub-ordinate-event.* Overall evaluation results of the systems are presented in Table 3. Results show that the system performs best with the feature template that represents the context of size five (i.e. previous two, current and the next two <*main-event*, *main-event*> pairs); tense and aspect of the current pair and the dynamic output relation of the previous <*main-event*, *main-event*> pair. It shows the precision, recall and F-score values of 43.8%, 43.8% and 43.8%, respectively under the strict evaluation scheme and 46.9%, 46.9% and 46.9%, respectively under the relaxed evaluation scheme.

For the *sub-ordinate-event* type relations, the system performs best with a feature template that corresponds to the context of size seven (i.e. previous three, current and the next three <*main-event*, *next-subordinate-event*, *previous- subordinate-event*, *main-event*> pairs); tense, aspect and class features of the current pair and the output relation of the previous pair, determined dynamically at run-time. It shows the precision, recall and F-score values of 55.1%, 55.1% and 55.1%, respectively under the strict evaluation scheme and 56.9%, 56.9% and 56.9%, respectively under the relaxed evaluation scheme.

**Table 3: Evaluation results of Task C (we report percentages)**

| Technique | Evaluation scheme | Precision | Recall | F-score |
|---|---|---|---|---|
| Baseline | Strict | 42.0 | 42.0 | 42.0 |
| | Relaxed | 46.0 | 46.0 | 46.0 |
| CRF (*main-event*) | Strict | 43.8 | 43.8 | 43.8 |
| | Relaxed | 46.9 | 46.9 | 46.9 |
| CRF(*subordinate-event* ) | Strict | 55.1 | 55.1 | 55.1 |
| | Relaxed | 56.9 | 56.9 | 56.9 |

It also shows the results of the *baseline* model. In case of task C, the most frequent temporal relation present in the training data is OVERLAP. For the *main-event*, CRF based system performs better than the *baseline* model with the margins of 1.8% F-score in the strict evaluation scheme and 0.9% F-score in the relaxed evaluation scheme. Results clearly show that CRF is most effective to handle the *subordinate-event*. It shows the overall performance improvement of 13.1% and 10.9% F-scores over the *baseline* model in the strict and relaxed evaluation scheme, respectively. The system also exhibits superior performance for the

*subordinate-event* over the *main-event* with more than 11.3% and 10% F-scores in the strict and relaxed evaluation schemes, respectively.

## 6 Conclusion

In this paper, we have reported our work on temporal information extraction under the TempEval 2007 evaluation exercise. We proposed the supervised systems based on CRF for solving the problems of all the three tasks, namely Tasks A, B and C of TempEval-2007. Each of the models was developed using only the features, available in the TimeBank corpus. Evaluation results yield the F-score values of 59.8%, 73.8% and 43.8% for Tasks A, B and C, respectively under the strict evaluation scheme. All these systems show the F-score values of 61.1%, 74.8% and 46.9% for Tasks A, B and C, respectively under the relaxed evaluation scheme. For the sub-event relation in Task C, the system showed the F-score values of 55.1% and 56.9% under the strict and relaxed evaluation scheme, respectively.

We would like to experiment by considering all and more variations of the available features. In future, we also want to introduce additional features that may be extracted from our existing tools. Some rules may be identified to make the system more robust. Future works also include investigating other statistical learning techniques like Maximum Entropy and Support Vector Machine for solving the problems.

## References

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., Pustejovsky, and J.: SemEval-2007 Task 15: TempEval Temporal Relation Identification. *Proceedings of the 4th International Workshop on Semantic Evaluations (semEval-2007)*, pp. 75-80, Prague.

Boguraev, B. and R. K. Ando. 2005. TimeMLCompliant Text Analysis for Temporal Reasoning. *Proceedings of Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, pp. 997–1003, Edinburgh, Scotland.

Mani, I., Wellner, B., Verhagen, M., Lee C.M., Pustejovsky, J. 2006. Machine Learning of Temporal Relation. *Proceedings of the 44th Annual meeting of the Association for Computational Linguistics*, pp. 753-760, Sydney, Australia.

Mani, I., B. Wellner, M. Verhagen, and J. Pustejovsky. 2007. Three Approaches to Learning TLINKs in TimeML. *Technical Report CS-07-268, Computer Science Department, Brandeis University*, Waltham, USA.

Chambers, N., S. Wang, and D. Jurafsky. 2007. Classifying Temporal Relations between Events. *Proceedings of the ACL 2007 Demo and Poster Sessions*, pp. 173–176, Prague, Czech Republic.

Pustejovsky, J., Castano, J., Ingria, R., Sauri, R.,Gaizauskas, R.,Setzer, A., Katz, G., Radev., D. 2003. TimeML: Robust Specification of Event and Temporal Expressions in Text. *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*, Tilburg.

Lafferty, J., McCallum, A., Pereira, F. 2001 Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of 18th International Conference on Machine Learning* (*ICML*), pp. 282-289.

Sha, F., Pereira, F. 2003. Shallow Parsing with Conditional Random Fields. *Proceedings of HLT-NAACL*, pp.134-141.