LCC-TE: A Hybrid Approach to Temporal Relation Identification in News Text

Congmin Min
Language Computer Corporation
1701 N. Collins Blvd. Suite 2000
Richardson, TX 75080
cmin@languagecomputer.com

Munirathnam Srikanth
Language Computer Corporation
1701 N. Collins Blvd, Suite 2000
Richardson, TX 75080
Srikanth.munirathnam
@languagecomputer.com

Abraham Fowler
Language Computer Corporation
1701 N. Collins Blvd, Suite 2000
Richardson, TX 75080
abraham@languagecomputer.com

Abstract

This paper explores a hybrid approach to temporal information extraction within the TimeML framework. Particularly, we focus on our initial efforts to apply machine learning techniques to identify temporal relations as defined in a constrained manner by the TempEval-2007 task. We explored several machine learning models and human rules to infer temporal relations based on the features available in TimeBank, as well as a number of other features extracted by our in-house tools. We participated in all three sub-tasks of the TempEval task in SemEval-2007 workshop and the evaluation shows that we achieved comparable results in Task A & B and competitive results in Task C.

1 Introduction

There has been a growing interest in temporal information extraction in recent years, as more and more operational NLP systems demands dealing with time-related issues in natural languages. In this paper, we report on an end-to-end system that is capable of automating identification of temporal referring expressions, events and temporal relations in text by leveraging various NLP tools and linguistic resources at LCC.

It has to be noted that the system we report here is not only intended for TempEval 2007 evaluation, but will also be used as a NLP tool for our other applications (e.g. temporal Question Answering). That is why we experimented to use our own temporal and event extraction capabilities in this work, although time and event tags have already been provided in the testing/training data. Another reason we use our own temporal tagging is that our temporal tagger extracts more information than that available in the training/testing data. For instance, temporal signals are removed from the data that the task organizers provide, but our temporal tagger detects that, as part of the tagging procedure. The following is an example for the tagged expression "on this coming Sunday".

```
<ArgStructure id="65" type="timex">
    <argRef type="determiner" tokStr="this"/>
     <argRef type="directionIndicator" tokStr="coming"/>
     <argRef type="focus" tokStr="Sunday"/>
     <argRef type="prepSignal" tokStr="on"/>
     <argRef type="head" tokStr="this coming Sunday"/>
     <argRef type="root" tokStr="on this coming Sunday"/>
    <argValue type="focusType" value="weekOfDay"/>
    <argValue type="subType" value="Fuzzy"/>
    <argValue type="type" value="Date"/>
</ArgStructure>
```

Our data structure allows us to easily access and manipulate any part of the tagged chunk of text, which leaves the interpretation of whether the temporal signal *on* in the example is part of the temporal expression to users of temporal tagger. Taking as input this data structure, the normalization, including relative date resolution, is a straightforward process, provided that the reference time can be computed from the context.

For temporal relation identification, by leveraging the capabilities of our temporal tagger, event tagger and several other in-house NLP tools, we derive a rich set of syntactic and semantic features for use by machine learning. We also explored the possibility of combining the rule-based approach with machine learning in an integrated manner so that our system can take advantage of these two approaches for temporal relation identification.

2 System Architecture

The overall architecture of our end-to-end system is illustrated in Figure 1 (Page 2).

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 219–222,
Prague, June 2007. ©2007 Association for Computational Linguistics

In addition to several common NLP tools, e.g. Named Entity Recognizer, we use syntactic and semantic parsers to identify syntactic and semantic roles (e.g. AGENT or SUBJECT) of event terms and a context detector to detect linguistic contexts in a discourse. We use such information as extended features for machine learning. The Temporal Tagger tags and normalizes temporal expressions conforming to the TimeML guideline. The Temporal Merger compares our own temporal and event tagging with those supplied in training/testing data. If there is any inconsistency, it will replace the former with the latter, which guarantees that our temporal and event tagging are the same as those in training/testing data. Feature Extractor extracts and composes features from documents processed by the NLP tools. Machine Learner and Human Rule Predictor take as input the feature vector for each instance to predict temporal relation. The Human Rule Predictor is a rule interpreter that read hand-crafted rules from plain text file to match each event instance represented by a feature vector.

Note that in Figure 1, *Syntactic Parsing* is done by a probabilistic chart parser, which generates full parse tree for each sentence. *Syntactic Pattern Matching* is performed by a syntactic pattern matcher, which operates on parse trees produced by chart parser and used by Temporal Tagger to tag and normalize temporal expressions.

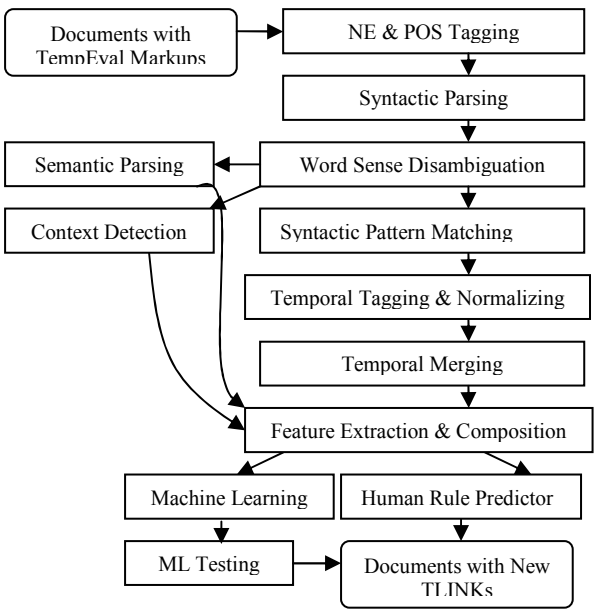

Figure 1. Overall System Architecture

3 Feature Engineering

While temporal tagging and normalization is rule-based in our system, temporal relation identification is a combination of machine learning and rule-based approaches. For machine learning, the feature set for the three tasks A, B and C we engineered consist of what we call 1) *first-class* features; 2) *derived* features; 3) *extended* features, and 4) *merged* features. The way we name the type of features is primarily for illustrating purpose.

3.1 First-class Features

The first-class features consist of:

- Event Class
- Event Stem
- Event and time strings
- Part of Speech of event terms
- Event Polarity
- Event Tense
- Event Aspect
- Type of temporal expression
- Value of temporal expression

The set of first-class features, which are directly obtained from the markups of training/testing data, are important, because most of them, including Event Class, Event Stem, POS, Tense and Type of Temporal Expression, have a great impact on performance of machine learning classifiers, compared with effects of other features.

3.2.2 Derived Features

From the first-class features, we derive and compute a number of other features:

- Tense and aspect shifts[1]
- Temporal Signal
- Whether an event is enclosed in quotes
- Whether an event has modals prior to it
- Temporal relation between the Document Creation Time and temporal expression in the target sentence.

The way we compute tense and aspect shifts is taking pair of contiguous events and assign a true/false value to each relation instance based on whether tense or shift change in this pair. Our experiments show that these two features didn't contribute to the overall score, probably because they are redundant with the Tense and Aspect features of each event term. Temporal Signal

[1] Initially used in (Mani, et. al. 2003)

represents temporal prepositions and they slightly contribute to the overall score of classifiers.

The last feature in this category is the Temporal Relation between the Document Creation Time and the Temporal Expression in the target sentence. The value of this feature could be "greater than", "less than", "equal", or "none". Experiments show that this is an important feature for Task A and B, because it contributes several points to the overall score. This value may be approximate for a number of reasons. For example, we can't directly compare a temporal expression of type Date with another expression of type Duration. However, even if we apply a simple algorithm to compute this relationship, it results in a noticeably positive effect on the performance of the classifier.

3.2.3 Extended Features

Features in the third category are extracted by our in-house tools, including:

- Whether an event term plays primary semantic or syntactic roles in a sentence
- Whether an event and a temporal expression are situated within the same linguistic context
- Whether two event terms co-refer in a discourse (This feature is only used for Task C)

Investigation reveals that different types of events defined in TimeML may or may not have specific semantic or syntactic roles (e.g. THM or OBJECT) in a particular context, therefore having an impact on their ways to convey temporal meanings. Experiments show that use of semantic and syntactic roles as binary features slightly increases performance.

The second feature in this category is *Context feature*. We use a context detection tool, which detects typical linguistic contexts, such as Reporting, Belief, Modal, etc. to decide whether an event and a temporal expression are within one context. For example[2],

- The company has **reported *declines*** *in operating profit in each of the past three years*, despite steady sales growth.

In this example, we identify a Reporting context with its signal *reported.* The temporal expression *each of the past three years* and the event *declines* are within the same context (the feature value would be *TRUE*). We intend this feature can help solve the problem of anchoring an event to its actual temporal expressions. In fact, we don't benefit from the use of this feature, probably because detecting those linguistic contexts is a problem in itself.

The third feature in this category is co-referential feature, which is only used for Task C. This feature indicates if two event terms within or outside one sentence are referring to the same event. Experiments show that this global feature produces a positive effect on the overall performance of the classifier.

3.2.4 Merged Features

The last type of feature we engineered is the *merged* feature. Due to time constraint, as well as the fact that the system for Task B produces better results than Task A and C, we only experimented merging the output of the system for Task B into the feature set of Task C and we achieved noticeable improvements because of adding this feature.

Most of the features introduced above are experimented in all three tasks A, B and C, except that the co-referential feature and the merged feature are only used in Task C. Also, in Task C since for each relation there are two events and possibly two temporal expressions, the number of features used is much more than that in Task A and B. The total number of features for Task C's training is 35 and 33 for testing.

3.1 Combination of Machine Learning and Human Rule

The design of our system allows both human rule-based and machine learning-based decision making. However, we have not decided exactly in what situations machine learning and human rule prediction should be used given a particular instance. The basic idea here is that we want to have the option to call either component on the fly in different situations so that we can take advantage of the two empirical approaches in an integrated way. We did some initial experiments on dynamically applying Human Rule Predictor and Machine Learner on Task B and we were able to obtain comparable results with or without using hand-crafted rules. As pointed out in (Li, et, al. 2006), Support Vector Machine, as well as other classifiers, makes most mistakes near the decision plane in feature space. We will investigate the

[2] This sentence is taken from the file *wsj_0027.tml* in TempEval 2007's training data.

possibility of applying human rule prediction to those relation instances where Machine Learning makes most mistakes.

3.2 Experiments and Results

Based on the features discussed in Section 3.3, we did a series of experiments for each task on four models: Naive-Bayes, Decision Tree (C5.0), Maximum Entropy and Support Vector Machine. Due to space constraint, we only report results from SVM model [3], which produces best performance in our case.

We here report two sets of performance numbers. The first set is based on our evaluation against a set of held-out data, 20 documents for each task, which were taken from the training data. The second set of performance numbers is based on evaluation against the final testing data provided by task organizers.

	strict			relaxed		
	P	R	F	P	R	F
Task A	0.68	0.68	0.68	0.69	0.69	0.69
Task B	0.80	0.80	0.80	0.82	0.82	0.82
Task C	0.63	0.63	0.63	0.67	0.67	0.67

Table 1. Performance figures evaluated against held-out data

	strict			relaxed		
	P	R	F	P	R	F
Task A	0.59	0.57	0.58	0.61	0.60	0.60
Task B	0.75	0.71	0.73	0.75	0.72	0.74
Task C	0.55	0.55	0.55	0.60	0.60	0.60

Table 2. Performance figures evaluated against testing data

Team	strict			relax		
	P	R	F	P	R	F
Ours	0.59	0.57	0.58	0.61	0.60	0.60
Average	0.59	0.54	0.56	0.62	0.57	0.59
Best	0.62	0.62	0.62	0.64	0.64	0.64

Table 3. Performance figures in Comparison for Task A

Team	strict			relax		
	P	R	F	P	R	F
Ours	0.75	0.71	0.73	0.76	0.72	0.74
Average	0.76	0.72	0.74	0.78	0.74	0.75
Best	0.80	0.80	0.80	0.84	0.81	0.81

Table 4. Performance figures in comparison for Task B

Team	strict			relax		
	P	R	F	P	R	F
Ours	0.55	0.55	0.55	0.60	0.60	0.60
Average	0.51	0.51	0.51	0.60	0.60	0.60
Best	0.55	0.55	0.55	0.66	0.66	0.66

Table 5. Performance figures in comparison for Task C

[3] We use the LIBSVM implementation of SVM, available at http://www.csie.ntu.edu.tw/cjlin/libsvm

According to Table 1 and 2, it appears that there are significant differences between the TLINK patterns in the held-out data and the final testing data, since the performance of the classifier shows an apparent discrepancy in two cases.

Table 3, 4 and 5 show performance numbers of our system, the average and the best system in comparison. There are six teams in total participating in the TempEval 2007 evaluation this year.

4 Conclusion

We participated in the SemEval2007 workshop and achieved encouraging results by devoting our initial efforts in this area. In next step, we plan to seek ways to expand the training data, implement quality human rules by performing rigorous data analysis, and explore use of more features for machine learning through feature engineering.

References

B. Boguraev and R.K. Ando. 2005. TimeML-compliant Text Analysis for Temporal Reasoning. *Proceedings of IJCAI, UK.*

D. Ahn, S.F. Adafre and M.D. Rijke. 2005. Towards Task-based Temporal Extraction and Recognition. *Dagstuhl Seminar Proceedings 05151.*

Inderjeet Mani and George Wilson. 2000. Robust Temporal Processing of News. *Proceedings of ACL'2000*.

Inderjeet Mani, Barry Schiffman, and Jianping Zhang. 2003. Inferring Temporal Ordering of Events in News. *Proceedings of HLT-NAACL'03*, 55-57.

K. Hacioglu, Y. Chen and B. Douglas. 2005. Automatic Time Expression Labeling for English and Chinese Text, P*roceedings of CICLing-2005.*

L. Li, T. Mao, D. Huang and Y. Yang. 2006. Hybrid Models for Chinese Named Entity Recognition. *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing.*

The TimeML Working Group. 2005. The TimeML 1.2 Specification. *http://www.timeml.org/site/publications/specs.html*