Temporal Relation Identification with Endpoints

Chong Min Lee
Georgetown University
3700 O Street, NW
Washington, D.C. 20057, USA
cml54@georgetown.edu

Abstract

Temporal relation classification task has issues of fourteen target relations, skewed distribution of the target relations, and relatively small amount of data. To overcome the issues, methods such as merging target relations and increasing data size with closure algorithm have been used. However, the method using merged relations has a problem on how to recover original relations. In this paper, a new reduced-relation method is proposed. The method decomposes a target relation into four pairs of endpoints with three target relations. After classifying a relation of each endpoint pair, four classified relations are combined into a relation of original fourteen target relations. In the combining step, two heuristics are examined.

1 Introduction

An interesting task in temporal information processing is how to identify a temporal relation between a pair of temporal entities such as events (EVENT) and time expressions (TIMEX) in a narrative. After the publication of TimeBank (Pustejovsky et al., 2003b) annotated in TimeML (Pustejovsky et al., 2003a), supervised learning techniques have been tested in the temporal relation identification task with different types of temporal entity pairs (Chambers et al., 2007; Boguraev and Ando, 2005; Verhagen et al., 2007).

There are three issues in applying supervised machine learning methods to this task. The first issue is that a temporal entity pair that is defined as a directed temporal link (TLINK) in TimeML should be classified into a relation among fourteen relations. The second issue is that the number of TLINKs is relatively small in spite of the fourteen targets. The third issue is skewed distributions of the relations. Without the solutions of the issues, it is impossible to achieve good performance in temporal relation identification through machine learning techniques.

Several solutions have been suggested such as increased number of TLINKs with a transitivity closure algorithm (Mani et al., 2007; Chambers et al., 2007) and decreased target relations into six (Mani et al., 2006; Chambers et al., 2007; Tatu and Srikanth, 2008) or three (Verhagen et al., 2007). An issue of the reduced-relation method is how to recover original relations. A module for the recovery can cause performance degeneration and seems intuitively inappropriate.

In this paper, a new reduced-relation method is presented. The method uses endpoints of temporal entities. A TimeML relation can be represented into four endpoint pairs with three relations: *before, equal,* and *after*. This method requires four relation identification classifiers among endpoints for a TLINK and each classifier has only three target relations instead of fourteen. The four classified relations need to be combined in order to restore an interval-based relation. In this study, the performance of the proposed method will be evaluated in identifying TLINK relations between temporal entities empirically.

Firstly, related studies are described in section 2. Secondly, the identification of four pointwise relations is described. Thirdly, methods for the combination of pointwise relations are explained. Finally,

Proceedings of the NAACL HLT 2010 Student Research Workshop, pages 40–45,
Los Angeles, California, June 2010.

the outlook of the proposed method is proposed.

2 Background

Temporal relation identification has three problems: sparse data, fourteen target relations, and skewed distribution. To reduce the problems, previous studies have used techniques such as increasing data size with closure algorithm and merging target relations.

Mani et al. (2006) used closure algorithm to increase training data size and merged inverse relations into six main relations. Their study applied the methods to classify relations of all TLINKs and showed the benefit of the methods in temporal relation identification. Chambers et al. (2007) reported 67.0% accuracy on the relation identification task among EVENT-EVENT (EE) TLINKs using the merged relations. And, the accuracy is the best performance with EE TLINKs.

The merging method assumes that target relations of TLINKs is already known. When a TLINK relation from an anchor to a target is *AFTER*, it can be changed into *BEFORE* by conversing the anchor and the target each other. When unknown instance is given, the merging process is impossible. When six merged relations were used as target relations, we assumes the conversion is already done. And the assumption is inappropriate.

TempEval07 (Verhagen et al., 2007) integrated 14 TLINK relations into three: *before*, *after*, and *overlap*. *overlap* is an extended relation that covers 12 relations except *BEFORE* and *AFTER*. This approach has a burden to recover 12 relations from the extensive one.

In this study, a TLINK is decomposed into four pairs of endpoint links in the step of applying machine learning approaches. Then, four classified endpoint relations are combined into a TimeML relation. Allen (1983) showed a relative order between intervals can be decomposed into relative orders of four endpoint pairs. In TimeML, temporal entities, EVENT and TIMEX, are intervals. An interval has a pair of endpoints: start and end. A relation between two intervals can be represented into relations of four pairs of starts and ends as in Table 2. A relative order between endpoints can be represented with three relations: *before*, *equal*, and *after*. The proposed method will be empirically investigated in this study.

3 Resources and Data Preparation

3.1 Temporal Corpora

TimeBank and Opinion corpora consist of 183 and 73 documents respectively. Among the documents, it is found that 42 documents have inconsistent TLINKs. The inconsistencies make it impossible to apply closure algorithm to the documents. Therefore, the 42 documents with inconsistent TLINKs are excluded. This study focuses on classifying relations of three types of TLINKs: TLINKs between EVENTs (EE), between an EVENT and a TIMEX (ET), and between an EVENT and Document Creation Time (ED).

As a preparation step, fourteen relations are merged into eleven relations (TimeML relations). *SIMULTANEOUS, IDENTITY, DURING,* and *DURUNG_BY* relations are identical in relative order between entities. Therfore, the relations are integrated into SIMULTANEOUS[1]. Then, closure algorithm is run on the documents to increase the number of TLINKs. The distribution of relations of three types is given in Table 1.

A document with merged relations is divided into four documents with endpoint relations: start of anchor and start of target, start of anchor and end of target, end of anchor and start of target, and end of anchor and end of target documents. The conversion table of a TimeML relation into four endpoint relations is given in Table 2 and the distribution of three relations after the conversion is given in 3.

4 Relation identification with end points

In endpoint relation identification experiment, support vector machine (SVM) and maximum entropy classifiers are built to classify three relations: *before, equal,* and *after*. First, feature vectors are constructed. When four endpoint links are from a TLINK, their feature vectors are identical except target endpoint relations.

[1] Mani et al. (2006) said DURING was merged into IS_INCLUSED. However, DURING, SIMULTANEOUS, and IDENTITY are converted into = of Allen's relations in `Tarski Toolkit` (Verhagen et al., 2005). In this paper, the implementation is followed.

Relation	EVENT-EVENT		EVENT-TIMEX		EVENT-DCT	
	Original	Closed	Original	Closed	Original	Closed
AFTER	735	11083	86	2016	169	259
BEFORE	1239	12445	160	1603	721	1291
BEGINS	35	75	23	36	0	0
BEGUN_BY	38	74	51	58	10	11
ENDS	15	64	65	128	0	0
ENDED_BY	87	132	43	61	6	6
IAFTER	38	138	3	8	1	1
IBEFORE	49	132	2	9	0	0
INCLUDES	246	3987	122	166	417	469
IS_INCLUDED	327	4360	1495	2741	435	467
SIMULTANEOUS	1370	2348	201	321	75	90

Table 1: Distribution of TimeML relations

TimeML Relation	**Inverse**	**Endpoint Relations**
x BEFORE y	y AFTER x	$x^- < y^-, x^- < y^+,$ $x^+ < y^-, x^+ < y^+$
x SIMULTANEOUS y	y SIMULTANEOUS x	$x^- = y^-, x^- < y^+,$ $x^+ > y^-, x^+ = y^+$
x IBEFORE y	y IAFTER x	$x^- < y^-, x^- < y^+,$ $x^+ = y^-, x^+ < y^+$
x BEGINS y	y BEGUN_BY x	$x^- = y^-, x^- < y^+,$ $x^+ > y^-, x^+ < y^+$
x ENDS y	y ENDED_BY x	$x^- > y^-, x^- < y^+,$ $x^+ > y^-, x^+ = y^+$
x INCLUDES y	y IS_INCLUDED x	$x^- < y^-, x^- < y^+,$ $x^+ > y^-, x^+ > y^+$

Table 2: Relation conversion table

End pairs	EVENT-EVENT			EVENT-TIMEX			EVENT-DCT		
	before	*equal*	*after*	*before*	*equal*	*after*	*before*	*equal*	*after*
start-start	1621 (39%)	1443 (35%)	1115 (27%)	327 (15%)	275 (12%)	1649 (73%)	1144 (62%)	85 (5%)	605 (33%)
start-end	3406 (82%)	38 (1%)	735 (18%)	2162 (**96%**)	3	86 (4%)	1664 (**91%**)	1	169 (9%)
end-start	1239 (30%)	49 (1%)	2891 (69%)	160 (7%)	2	2089 (**93%**)	721 (39%)	0	1113 (61%)
end-end	1650 (39%)	1472 (35%)	1057 (25%)	1680 (75%)	309 (14%)	262 (12%)	1156 (63%)	81 (4%)	597 (33%)

Table 3: Distribution of end point relations.

10-fold cross validation is applied at document-level. In some previous studies, all temporal links were collected into a set and the set was split into training and test data without the distinction on sources. However, the approach could boost system performance as shown in Tatu and Srikanth (2008).

When TLINKs in a file are split in training and test data, links in training data can be composed of similar words in test data. In that case, the links in training can play a role of background knowledge. Therefore, document-level 10-fold cross validation is exploited.

4.1 Features

In constructing feature vectors of three TLINK types, features that were used in order to identify TimeML relations in previous studies are adopted. The features have been proved useful in identifying a TimeML relation in the studies. Moreover, the features still seem helpful for endpoint relation identification task. For example, *past* and *present* tenses of two EVENTs could be a clue to make a prediction that *present* tensed EVENT is probably after *past* tensed EVENT.

Annotated information of EVENT and TIMEX in the temporal corpora is used in the feature vector construction. This proposed approach to use endpoint conversion in relation identification task is the first attempt. Therefore, the annotated values are used as features in order to see the effect of this approach. However, state-of-the-arts natural language processing programs such as Charniak parser and Porter Stemmer are sometimes used to extract additional features such as stems of event words, the existence of both entities in the same phrase, and etc.

> The company has ***reported declines*** in operating profit in ***the past three years***

Features for EVENT TENSE, ASPECT, MODAL, POS, and CLASS annotations are borrowed from temporal corpora as features. And, a stem of an EVENT word is added as a feature instead of a word itself in order to normalize it. *reported* is represented as <(TENSE:present), (ASPECT:perferce), (MODAL:none), (POS: verb), (CLASS: reporting), (STEM:report)>.

Features for TIMEX In the extraction of TIMEX features, it tries to capture if specific words are in a time expression to normalize temporal expressions. The time point of an expression can be inferred through the specific words such as *ago, coming, current, earlier* and etc. Additionally, the existence of plural words such as *seconds, minutes, hours, days, months,* and *years* is added as a feature. The specific words are:

- *ago, coming, current, currently, earlier, early, every, following, future, last, later, latest, next, now, once, past, previously, recent, recently, soon, that, the, then, these, this, today, tomorrow, within, yesterday,* and *yet*

the past three years are represented as <(AGO:0), (COMING:0), (CURRENT:0), (CURRENTLY:0), (EARLIER:0), (EARLY:0), (EVERY:0), (FOLLOWING:0), (FUTURE:0), (LAST:1), (LATER:0), (LASTED:0), (NEXT:0), (NOW:0), (ONCE:0), (PAST:1), (PREVIOUSLY:0), (RECENT:0), (RECENTLY:0), (SOON:0), (THAT:0), (THE:1), (THEN:0), (THESE:0), (THIS:0), (TODAY:0), (TOMORRWO:0), (WITHIN:0), (YESTERDAY:0), (YET:0), (PLURAL:1)>.

Relational features between entities In addition, relational information between two entities is used as features. It is represented if two entities are in the same sentence. To get the other relational information, a sentence is parsed with Charniak parser. Syntactic path from an anchor to a target is calculated from the parsed tree. A syntactic path from *reported* to *the past three years* is "VBN‖VP‖PP‖NP". It is represented if two entities are in the same phrase and clause with the path. When only one clause or phrase exists in the path except part-of-speeches of both entities, the features are marked as 1s. The counts of words, phrases, and clauses between temporal entities are also used as features. When two entities are not in the same sentence, 0s are given as the values of the features except the word count. Some prepositions and conjunctions are used as features when the words are used as a head word of syntactic path from an entity to the other entity. In the example of "VBN‖VP‖PP‖NP", "in" in "in the past three years" is the head word of PP. So, *in* is marked 1. The head words that are used as features are:

- *after, as, at, before, between, by, during, for, in, once, on, over, since, then, through, throughout, until, when,* and *while*

EE and ET types have feature vectors that consist of features of both entities and relational features. ED type has only features of EVENT.

5 Restoration of original relations

Four endpoint relations of a TLINK are classified in the previous section. The combination of the classified relations needs to be restored into a relation among the eleven merged TimeML relations. However, due to the independence of four classifiers, it is not guaranteed that a TimeML relation can be generated from four endpoint relations. When the restoration fails, the existence of errors in the four predictions is implied. In this step, two methods to restore a TimeML relation are investigated: Minimum Edit Distance (MED) and Highest Score (HS).

MED checks how many substitutions are needed to restore a TimeML relation. A TimeML relation with the minimum changes is defined as the restored relation. Let's suppose four endpoint relations are given such as x^- *before* y^-, x^- *after* y^+, x^+ *before* y^-, and x^+ *before* y^+. Among other possible ways to get a TimeML relation, *BEFORE* could be recovered with a change of *before* in x^- *after* y^+ into *before*. Therefore, *BEFORE* is chosen as a restored TimeML relation. When several candidates are available, a method is examined in selecting one. The method is to give weight on classifiers that show better performance. If two candidates are available by changing *before* of start-start or *before* of start-end in ET type, this method selects a candidate by changing *before* when *before* of start-end shows better performance.

HS uses the sum of confidence scores from classifiers. Each classifier of the four endpoint pairs generates confidence scores of three relations (*before*, *equal*, and *after*). Among 81 possible combinations of four classifiers with three target relations, the highest-scored one that can be restored into a TimeML relation is chosen as a prediction. When several candidates exist, the selection method of MED is also adopted.

6 Expectations and future plans

First, I will show how beneficial four endpoint systems are at identifying endpoint relations. F-measure will be used to show the performance of an endpoint relation classifier in identifying each endpoint relation. And, accuracy is used to report overall performance of the classifier. Second, I will show how effective the endpoint method is in identifying TLINK relations. I will build a base classifier with eleven TimeML relations and feature vectors that are identical with the endpoint systems. The performance difference in identifying TimeML relations between this proposed system and the base system will be presented to show whether this proposed approach is successful.

Previous research such as Verhagen et al. (2007) using three reltions as target relations showed from 60% to 80% performance according to TLINK types. Moreover, some distributions of endpoint relations show over 90% such as *before* of end-start in ET and ED TLINKs, and *after* of end-start in ET TLINK in Table 3. Therefore, we can expect each endpoint identification system will perform well in classifying endpoint relations.

The success of this new approach will depend on the restoration step. The excessively skewed distributions can make similar predicted sequences of endpoint relations. It can weaken the advantage of this endpoint approach that every TimeML relation can be generated through combining endpoint relations. For example, *equal* shows very small distributions in start-end and end-start endpoint pairs. Therefore, it is probable that TimeML relations such as *IAFTER* and *IBEFORE* cannot be classified correctly. It can be a challenge how to correctly classify endpoint relations with small distribution.

One possible solution for the challenge is to check global consistency among classified relations such as Bramsen et al. (2006) and Chambers and Jurafsky (2008). The global consistency restoration can give a chance to replace excessively distributed relations with sparse relations. However, *equal* is used additionally in this study. Therefore, modifications in the method of Bramsen et al. (2006) and Chambers and Jurafsky (2008) are needed before applying their method.

References

James Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the Association for Computing Machinery*, 26(1):832–843.

Branimir Boguraev and Rie Kubota Ando. 2005. TimeML-compliant text analysis for temporal reasoning. In *Proceedings of the 2005 International Joint Conference on Artificial Intelligence*, pages 997–1003.

Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006. Inducing temporal graphs. In *Proceedings of the 2006 Conference on Empirical Methods on Natural Language Processing*, pages 189–198.

Nathanael Chambers and Dan Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 698–706, Morristown, NJ, USA. Association for Computational Linguistics.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of 45th Annual Meeting of the Association for Computational Linguistics*, pages 173–176.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

Inderjeet Mani, Ben Wellner, Marc Verhagen, and James Pustejovsky. 2007. Three approaches to learning tlinks in timeml. Technical Report CS-07-268, Brandeis University, Waltham, MA, USA.

James Pustejovsky, Jose Castao, Robert Ingria, Roser Saur, Robert Gaizauskas, and Andrea Setzer. 2003a. TimeML: robust specification of event and temporal expressions in text. In *IWCS-5 Fifth International Workshop on Computational Semantics*.

James Pustejovsky, Patrick Hanks, Roser Saur, Andrew See, David Day, Lisa Ferro, Robert Gaizauskas, Marcia Lazo, Andrea Setzer, and Beth Sundheim. 2003b. The TimeBank corpus. In *Proceedings of Corpus Linguistics 2003*, pages 647–656, Lancaster, UK.

Marta Tatu and Munirathnam Srikanth. 2008. Experiments with reasoning for temporal relations between events. In *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*, pages 857–864, Morristown, NJ, USA. Association for Computational Linguistics.

Marc Verhagen, Inderjeet Mani, Roser Sauri, Robert Knippen, Seok Bae Jang, Jessica Littman, Anna Rumshisky, John Phillips, and James Pustejovsky. 2005. Automating temporal annotation with tarsqi. In *ACL '05: Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 81–84, Morristown, NJ, USA. Association for Computational Linguistics.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague.