Exploiting Timegraphs in Temporal Relation Classification

Natsuda Laokulrat[†], Makoto Miwa[‡], and Yoshimasa Tsuruoka[†]

[†]The University of Tokyo, 3-7-1 Hongo, Bunkyo-ku, Tokyo, Japan
{natsuda,tsuruoka}@logos.t.u-tokyo.ac.jp
[‡]Toyota Technological Institute, 2-12-1 Hisakata, Tempaku-ku, Nagoya, Japan
miwa@toyota-ti.ac.jp

Abstract

Most of the recent work on machine learning-based temporal relation classification has been done by considering only a given pair of temporal entities (events or temporal expressions) at a time. Entities that have temporal connections to the pair of temporal entities under inspection are not considered even though they provide valuable clues to the prediction. In this paper, we present a new approach for exploiting knowledge obtained from nearby entities by making use of timegraphs and applying the stacked learning method to the temporal relation classification task. By performing 10-fold cross validation on the Timebank corpus, we achieved an F1 score of 59.61% based on the graph-based evaluation, which is 0.16 percentage points higher than that of the local approach. Our system outperformed the state-of-the-art system that utilizes global information and achieved about 1.4 percentage points higher accuracy.

1 Introduction

Temporal relationships between entities, namely temporal expressions and events, are regarded as important information for deep understanding of documents. Being able to predict temporal relations between events and temporal expressions within a piece of text can support various NLP applications such as textual entailment (Bos et al., 2005), multi-document summarization (Bollegala et al., 2010), and question answering (Ravichandran and Hovy, 2002).

Temporal relation classification, which is one of the subtasks TempEval-3 (UzZaman et al., 2013), aims to classify temporal relationships between pairs of temporal entities into one of the 14 relation types according to the TimeML specification (Pustejovsky et al., 2005), e.g., *BEFORE, AFTER, DURING,* and *BEGINS.*

The Timebank corpus introduced by Pustejovsky et al. (2003) has enabled the machine learning-based classification of temporal relationship. By learning from the annotated relation types in the documents, it is possible to predict the temporal relation of a given pair of temporal entities (Mani et al., 2006).

However, most of the existing machine learning-based systems use local information alone, i.e., they consider only a given pair of temporal entities at a time. Entities that have temporal connections to the entities in the given pair are not considered at all even though they provide valuable clues to the prediction. Hence, the local approach often produces contradictions. For instance, the system may predict that A happens before B, that B happens before C, and that A happens after C, which are mutually contradictory.

In order to tackle the contradiction problem, global approaches have been proposed by Chambers and Jurafsky (2008) and Yoshikawa et al. (2009). Chamber and Jurafsky proposed a global model based on Integer Linear Programming that combines the output of local classifiers and maximizes the global confidence scores. While they focused only on the temporal relations between events, Yoshikawa et al. proposed a Markov Logic model to jointly predict the temporal relations between events and time expressions.

In this paper, we propose an approach that utilizes timegraphs (Miller and Schubert, 1999), which represent temporal connectivity of all temporal entities in each document, for the relation classification. Our method differs from the previous work in that their methods used transition rules to enforce consistency within each triplet of relations, but our method can also work with a set consisting of more than three relations. Moreover,

Proceedings of TextGraphs-9: the workshop on Graph-based Methods for Natural Language Processing, pages 6–14,
October 29, 2014, Doha, Qatar.

In <TIMEX3 tid=“t88” type=“DURATION” value=“P9M” temporalFunction=“true” functionInDocument=“NONE” endPoint=“t0”>**the first nine months**</TIMEX3>**, profit** <EVENT eid=“e30” class=“OCCURRENCE”>**rose**</EVENT> **10% to $313.2 million, or $3.89 a share, from $283.9 million, or $3.53 a share.**

<MAKEINSTANCE eventID=“e30” eiid=“ei349” tense=“PAST” aspect=“NONE” polarity=“POS” pos=“VERB” />

<TLINK lid=“l23” relType=“DURING” eventInstanceID=“ei349” relatedToTime=“t88” />

Figure 1: An example from the Timebank corpus

in our work, the full set of temporal relations specified in TimeML are used, rather than the reduced set used in the previous work.

We evaluate our method on the TempEval-3's Task C-relation-only data, which provides a system with all the appropriate temporal links and only needs the system to classify the relation types. The result shows that by exploiting the timegraph features in the stacked learning approach, the classification performance improves significantly. By performing 10-fold cross validation on the Timebank corpus, we can achieve an F1 score of 59.61% based on the graph-based evaluation, which is 0.16 percentage points (*pp*) higher than that of the local approach. We compared the results of our system to those of Yoshikawa et al. (2009) and achieved about 1.4 *pp* higher accuracy.

The remainder of the paper is organized as follows. Section 2 explains the temporal relation classification task and the pairwise classifier. Section 3 and Section 4 describe our proposed timegraph features and the application to the stacked learning approach. Section 5 shows the experiment setup and presents the results. Finally, we discuss the results in 6 and conclude with directions for future work in Section 7.

2 Temporal Relation Classification

According to TempEval-3, a temporal annotation task consists of several subtasks, including temporal expression extraction (Task A), event extraction (Task B), and temporal link identification and relation classification (Task C). Our work, as with the previous work mentioned in Section 1, only focuses on the relation classification task (Task C-relation only). The system does not extract events and temporal expressions automatically.

A pair of temporal entities, including events and temporal expressions, that is annotated as a temporal relation is called a TLINK. Temporal relation classification is a task to classify TLINKs into temporal relation types.

Following TempEval-3, all possible TLINKs are between:

- Event and Document Creation Time (DCT)
- Events in the same sentence
- Event and temporal expression in the same sentence
- Events in consecutive sentences

2.1 The Timebank corpus

The Timebank corpus is a human-annotated corpus commonly used in training and evaluating a temporal relation classifier. It is annotated following the TimeML specification to indicate events, temporal expressions, and temporal relations. It also provides five attributes, namely, *class, tense, aspect, modality,* and *polarity*, associated with each event *(EVENT)*, and four attributes, namely, *type, value, functionInDocument,* and *temporalFunction*, associated with each temporal expression *(TIMEX3)*. An example of the annotated event and temporal expression is shown in Figure 1. The sentence is brought from wsj_0292.tml in the Timebank corpus.

There is no modal word in the sentence, so the attribute *modality* does not appear.

We use the complete set of the TimeML relations, which has 14 types of temporal relations including *BEFORE, AFTER, IMMEDIATELY BEFORE, IMMEDIATELY AFTER, INCLUDES, IS INCLUDED, DURING, DURING INVERSE, SIMULTANEOUS, IDENTITY, BEGINS, BEGUN BY, END,* and *ENDED BY.* However, in TempEval-3, *SIMULTANEOUS* and *IDENTITY* are regarded as the same relation type, so we change all *IDENTITY* relations into *SIMULTANEOUS*.

Given the example mentioned above, the temporal relation is annotated as shown in the last line of Figure 1. From the annotated relation, the event **rose (e30)** happens *DURING* the temporal expression **the first nine months (t88)**.

Feature	E-E	E-T	Description
Event attributes			
Class	X	X	All attributes associated with events. The explanation of each attribute can be found in (Pustejovsky et al., 2005).
Tense	X	X	
Aspect	X	X	
Modality	X	X	
Polarity	X	X	
Timex attributes			
Type		X	All attributes associated with temporal expressions. The explanation of each attribute can be found in (Pustejovsky et al., 2005).
Value		X	
FunctionInDocument		X	
TemporalFunction		X	
Morphosyntactic information			
Words	X	X	Words, POS, lemmas within a window before/after event words extracted using Stanford coreNLP (Stanford NLP Group, 2012)
Part of speech tags	X	X	
Lemmas	X	X	
Lexical semantic information			
Synonyms of event word tokens	X	X	WordNet lexical database (Fellbaum, 1998)
Synonyms of temporal expressions		X	
Event-Event information			
Class match	X		Details are described in (Chambers et al., 2007)
Tense match	X		
Aspect match	X		
Class bigram	X		
Tense bigram	X		
Aspect bigram	X		
Same sentence	X	X	True if both temporal entities are in the same sentence
Deep syntactic information			
Phrase structure	X	X	Deep syntactic information extracted from Enju Parser (Miyao and Tsujii, 2008). The details are described in (Laokulrat et al., 2013)
Predicate-argument structure	X	X	

Table 1: Local features

Feature	E-E	E-T	Description
Adjacent nodes and links	X	X	The details are described in Subsection 3.2
Other paths	X	X	
Generalized paths	X	X	
(E,V,E) tuples	X	X	
(V,E,V) tuples	X	X	

Table 2: Timegraph features

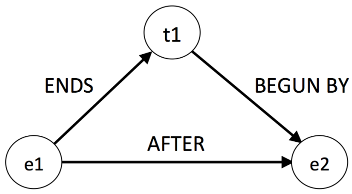


Figure 2: path length ≤ 2

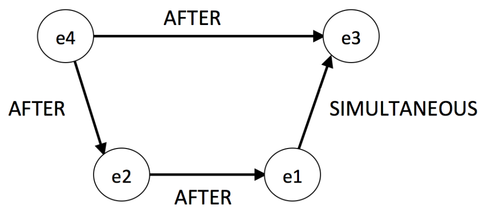


Figure 3: path length ≤ 3

3 Proposed method

Rather than using only local information on two entities in a TLINK, our goal is to exploit more global information which can be extracted from a document's timegraph. Our motivation is that temporal relations of nearby TLINKs in a timegraph provide very useful information for predicting the relation type of a given TLINK. For instance, consider the following sentence and the temporal connectivity shown in Figure 2.

> *About 500 people* ***attended (e1)*** *a Sunday night memorial for the Buffalo-area physician who performed abortions,* ***one year (t1)*** *after he was* ***killed (e2)*** *by a sniper's bullet.*

It can be seen that the relation between ***e1*** and ***t1*** and the relation between ***t1*** and ***e2*** are useful for predicting the relation between ***e1*** and ***e2***.

Another more-complicated example is shown below with temporal connectivity in Figure 3.

> *"The Congress of the United States is* ***affording(e1)*** *Elian Gonzalez what INS and this administration has not, which is his legal right and his right to due process,"* ***said(e2)*** *Jorge Mas Santos, chairman of the Cuban American National Foundation. "This* ***gives(e3)*** *him the protection that he will not be* ***repatriated(e4)*** *to Cuba between now and Feb. 10."*

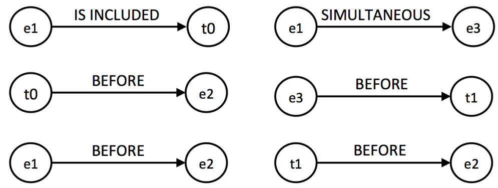


Figure 5: Local pairwise classification. Each TLINK is classified separately.

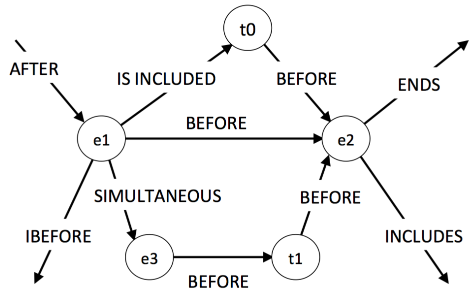


Figure 6: Timegraph constructed from a document's TLINKs

Again, the relation between ***e4*** and ***e3*** can be inferred from the nearby relations, i.e., (1) ***e4_AFTER_e2*** and ***e2_AFTER_e1*** imply ***e4_AFTER_e1***, (2) ***e4_AFTER_e1*** and ***e1_SIMULTANEOUS_e3*** imply ***e4_AFTER_e3***.

3.1 Overview of our framework

Our framework is based on the stacked learning method (Wolpert, 1992), which employs two stages of classification as illustrated in Figure 4.

3.1.1 Local pairwise model

In a local pairwise model, temporal relation classification is done by considering only a given pair of temporal entities at a time as illustrated in Figure 5. We use a supervised machine learning approach and employ the basic feature set that can be easily extracted from the document's text and the set of features proposed in our previous work (Laokulrat et al., 2013), which utilizes deep syntactic information, as baselines. The local features at different linguistic levels are listed in Table 1.

Two classifiers are used: one for Event-Event TLINKs (E-E), and the other for Event-Time TLINKs (E-T).

3.1.2 Stacked learning

Stacked learning is a machine learning method that enables the learner to be aware of the labels of nearby examples.

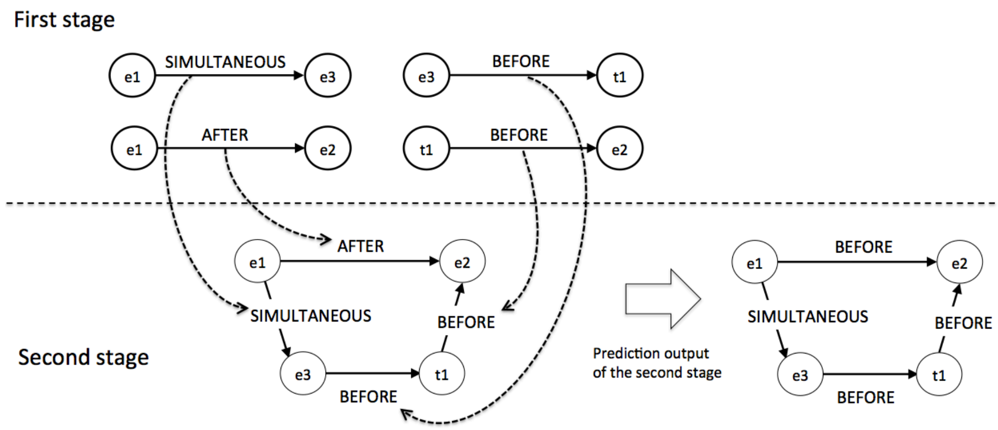


Figure 4: Stacked learning. The output from the first stage is treated as features for the second stage. The final output is predicted using label information of nearby TLINKs.

The first stage, as shown in Figure 5, uses the local classifiers and predicts the relation types of all TLINKs. In the second stage, the document's timegraph is constructed and the output from the first stage is associated with TLINKs in the graph. The classifiers in the second stage use the information from the nearby TLINKs and predict the final output. We exploit features extracted from the documents' timegraphs, as listed in Section 3.2 in the second stage of the stacked learning.

An example of a document's timegraph is shown in Figure 6.

3.2 Timegraph features

We treat timegraphs as directed graphs and double the number of edges by adding new edges with opposite relation types/directions to every existing edge. For example, if the graph contains an edge ***e1*** *_BEFORE_* ***e2***, we add a new edge ***e2*** *_AFTER_* ***e1***.

Our proposed timegraph features are described below.

- Adjacent nodes and links

  The features are the concatenation of the directions to the adjacent links to the pair of entities, the relation types of the links, and the information on the adjacent nodes, i.e., word tokens, part of speech tags, lemmas. For example, the features for predicting the relation between ***e1*** and ***e2*** in Figure 6 are *SRC_OUT-IS_INCLUDED*-(Type of ***t0***), *DEST_IN-BEFORE*-(Type of ***t0***), and so on.

  In this work, only Type of temporal expression (an attribute given in the Timebank corpus), Tense and Part-of-speech tag are applied but other attributes could also be used.

- Other paths

  Paths with certain path lengths (in this work, $2 \leq$ path length ≤ 4) between the temporal entities are used as features. The paths must not contain cycles. For example, the path features of the relation between ***e1*** and ***e2*** are *IS_INCLUDED-BEFORE* and *SIMULTANEOUS-BEFORE-BEFORE*.

- Generalized paths

  A generalized version of the path features, e.g., the *IS_INCLUDED-BEFORE* path is generalized to **-BEFORE* and *IS_INCLUDED-**.

- (E,V,E) tuples

  The (E,V,E) tuples of the edges and vertices on the path are used as features, e.g., *IS_INCLUDED_*(Type of ***t0***)*_BEFORE*.

- (V,E,V) tuples

  The (V,E,V) tuples of the edges and vertices on the path are used as features, e.g., (Tense of ***e1***)*_IS INCLUDED_*(Type of ***t0***) and (Type of ***t0***)*_BEFORE_*(Tense of ***e2***).

The summary of the timegraph features is shown in Table 2.

4 Relation inference and time-time connection

We call TLINKs that have more than one path between the temporal entities *"multi-path TLINKs"*. The coverage of the multi-path TLINKs is presented in Table 3. The annotated entities in the Timebank corpus create loosely connected timegraphs as we can see from the table that only 5.65% of all the annotated TLINKs have multiple paths between given pairs of temporal entities.

Since most of the timegraph features are only applicable for multi-path TLINKs, it is important to have dense timegraphs. In order to increase the numbers of connections, we employ two approaches: relation inference and time-time connection.

4.1 Relation inference

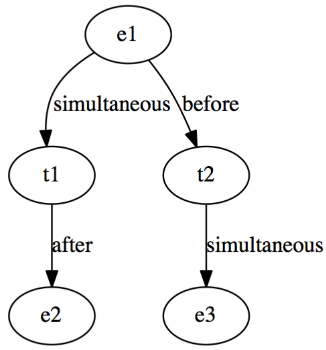
We create new E-E and E-T connections between entities in a timegraph by following a set of inference rules. For example, if ***e1*** happens *AFTER* ***e2*** and ***e2*** happens *IMMEDIATELY_AFTER* ***e3***, then we infer a new temporal relation "***e1*** happens *AFTER* ***e3***". In this paper, we add a new connection only when the inference gives only one type of temporal relation as a result from the relation inference. Figure 7b shows the timegraph after adding new inference relations to the original timegraph in Figure 7a.

4.2 Time-time connection

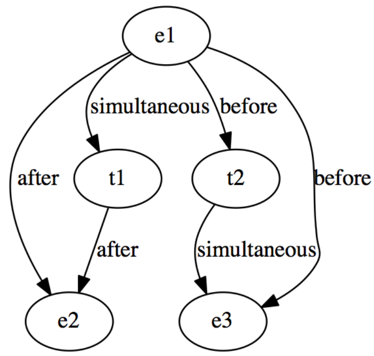
As with Chambers et al. (2007) and Tatu and Srikanth (2008), we also create new connections between time entities in a timegraph by applying some rules to normalized values of time entities provided in the corpus.

Figure 7c shows the timegraph after adding a time-time link and new inference relations to the original timegraph in Figure 7a. When the normalized value of ***t2*** is more than the value of ***t1***, a TLINK with the relation type *AFTER* is added between them. After that, as introduced in Subsection 4.2, new inference relations (***e1***-***e2***, ***e1***-***e3***, ***e2***-***e3***) are added.

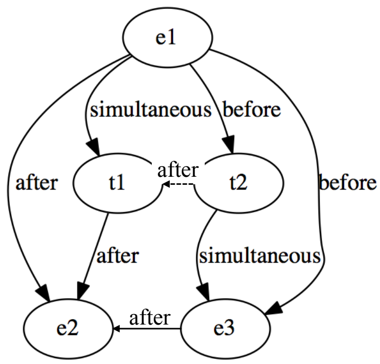
As the number of relations grows too large after performing time-time connection and inference relation recursively, we limited the number of TLINKs for each document's timegraph to 10,000 relations. The total number of TLINKs for all documents in the corpus is presented in Table 4. The first row is the number of the human-annotated relations. The second and third rows show the total number after performing relation inference and time-time connection.



(a) Original timegraph



(b) After relation inference. Two relations (e1-e2, e1-e3) are added.



(c) After time-time connection (t1-t2) and relation inference. Three relations (e1-e2, e1-e3, e2-e3) are added.

Figure 7: Increasing number of TLINKs

No. of TLINKs	E-E	E-T	Total
All TLINKs	2,520	2,463	4,983
Multi-path TLINKs	119	163	282
Percentage	4.72	6.62	5.65

Table 3: Coverage of multi-path TLINKs

Approach	Graph-based evaluation		
	F1(%)	P(%)	R(%)
Local - baseline features	58.15	58.17	58.13
Local - baseline + deep features	59.45	59.48	59.42
Stacked - baseline features	58.33	58.37	58.29
Stacked (inference) - baseline features	58.30	58.32	58.27
Stacked (inference, time-time) - baseline features	58.29	58.31	58.27
Stacked - baseline + deep features	59.55	59.51	59.58
Stacked (inference) - baseline + deep features	59.55	59.57	59.52
Stacked (inference, time-time) - baseline + deep features	**59.61**	**59.63**	**59.58**

Table 5: Ten-fold cross validation results on the training set

No. of TLINKs	Total
Annotated	4,983
+Inference	24,788
+Inference + time-time connection	87,992

Table 4: Number of TLINKs in the Timebank corpus

5 Evaluation

For the baselines and both stages of the stacked learning, we have used the LIBLINEAR (Fan et al., 2008) and configured it to work as L2-regularized logistic regression classifiers.

We trained our models on the Timebank corpus, introduced in Subsection 2.1, which was provided by the TempEval-3 organiser. The corpus contains 183 newswire articles in total.

5.1 Results on the training data

The performance analysis is performed based on 10-fold cross validation over the training data. The classification F1 score improves by 0.18 *pp* and 0.16 *pp* compared to the local pairwise models with/without deep syntactic features.

We evaluated the system using a graph-based evaluation metric proposed by UzZaman and Allen (2011). Table 5 shows the classification accuracy over the training set using graph-based evaluation.

The stacked model affected the relation classification output of the local model, changing the relation types of 390 (out of 2520) E-E TLINKs and 169 (out of 2463) E-T TLINKs.

5.2 Comparison with the state of the art

We compared our system to that of Yoshikawa et al. (2009) which uses global information to improve the accuracy of temporal relation classification. Their system was evaluated based on TempEval-2's rules and data set (Verhagen et al., 2007), in which the relation types were reduced to six relations: *BEFORE*, *OVERLAP*, *AFTER*, *BEFORE-OR-OVERLAP*, *OVERLAP-OR-AFTER*, and *VAGUE*. The evaluation was done using 10-fold cross validation over the same data set as that of their reported results.

According to TempEval-2's rules, there are three tasks as follows:

- Task A: Temporal relations between events and all time expressions appearing in the same sentence.
- Task B: Temporal relations between events and the DCT.
- Task C: Temporal relations betweeen main verbs of adjacent sentences.

The number of TLINKs annotated by the organizer, after relation inference, and after time-time connection for each task is summarized in Table 7. Table 8 shows the number of TLINKs after performing relation inference and time-time connection.

As shown in Table 6, our system can achieve better results in task B and C even without deep syntactic features but performs worse than their system in task A. Compared to the baselines, the overall improvement is statistically significant* (p $< 10^{-4}$, McNemar's test, two-tailed) without deep syntactic features and gets more statistically significant** (p $< 10^{-5}$, McNemar's test, two-tailed) when applying deep syntactic information to the system. The overall result has about 1.4 *pp* higher accuracy than the result from their global model. Note that Yoshikawa et al. (2009) did not apply deep syntactic features in their system.

Approach	Task A	Task B	Task C	Overall
Yoshikawa et al. (2009) (local)	61.3	78.9	53.3	66.7
Yoshikawa et al. (2009) (global)	66.2	79.9	55.2	68.9
Our system (local) - baseline features	59.9	80.3	58.5	68.5
Our system (local) - baseline + deep features	62.1	80.3	58.4	69.0
Our system (stacked) - baseline features	59.5	79.9	58.5	68.2
Our system (stacked, inference) - baseline features	59.9	80.0	**59.7**	68.7
Our system (stacked, inference, time-time) - baseline features	63.8	80.0	58.9	69.5*
Our system (stacked) - baseline + deep features	63.5	79.4	58.0	68.9
Our system (stacked, inference) - baseline + deep features	63.7	80.3	59.2	69.7
Our system (stacked, inference, time-time) - baseline + deep features	**65.9**	**80.5**	58.9	**70.3****

Table 6: Comparison of the stacked model to the state of the art and to our local model (F1 score(%))

No. of TLINKs	Task A	Task B	Task C
Annotated	1,490	2,556	1,744

Table 7: TempEval-2 data set

No. of TLINKs	Total
Annotated	5,970
+Inference	156,654
+Inference + time-time connection	167,875

Table 8: Number of relations in TempEval-2 data set

The stacked model enhances the classification accuracy of task A when timegraphs are dense enough. Deep syntactic features can be extracted only when temporal entities are in the same sentences so they improve the model for task A (event-time pairs in the same sentences) but these features clearly lower the accuracy of task C, since there are very few event-event pairs that appear in the same sentences (and break the definition of task C). This is probably because the sparseness of the deep features degrades the performance in task C. Moreover, these features do not help task B in the local model because we cannot extract any deep syntactic features from TLINKs between events and DCT. However, they contribute slightly to the improvement in the stacked model since deep syntactic features increase the accuracy of the prediction of task A in the first stage of the stacked model. As a result, timegraph features extracted from the output of the first stage are better than those extracted from the local model trained on only baseline features.

6 Discussion

As we can see from Table 5 and 6, although deep syntactic features can improve the classification accuracy significantly, some additional preprocessing is required. Moreover, deep parsers are not able to parse sentences in some specific domains. Thus, sometimes it is not practical to use this kind of features in real-world temporal relation classification problems. By applying the stacked learning approach to the temporal relation classification task, the system with only baseline features is able to achieve good classification results compared to the system with deep syntactic features.

Again, from Table 5 and 6, the inference and time-time connection, described in Section 4, sometimes degrade the performance. This is presumably because the number of features increases severely as the number of TLINKs increased.

The stacked model also has another advantage that it is easy to build and does not consume too much training time compared to MLNs used by Yoshikawa et al. (2009), which are, in general, computationally expensive and infeasible for large training sets.

7 Conclusion

In this paper, we present an approach for exploiting timegraph features in the temporal relation classification task. We employ the stacked learning approach to make use of information obtained from nearby entities in timegraphs. The results show that our system can outperform the state-of-the-art system and achieve good accuracy by using only baseline features. We also apply the relation inference rules and the time-time connection to tackle the timegraphs' sparseness problem.

In future work, we hope to improve the classification performance by making use of probability values of prediction results obtained from the first stage of the stacked learning and applying the full set of inference relations to the system.

Acknowledgement

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions, which were helpful in improving the quality of the paper.

References

Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. 2010. A bottom-up approach to sentence ordering for multi-document summarization. In *Information Processing & Management*, Volume 46, Issue 1, January 2010, pages 89–109.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *HLT/EMNLP 2005*, pages 628–635.

Nathanael Chambers, Shan Wang and Dan Jurafsky. 2007. Classifying temporal relations between events. In *ACL 2007*, pages 173–176.

Nathanael Chambers and Dan Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In *EMNLP 2008*, pages 698–706.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification.

Christiane Fellbaum. 1998. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

Natsuda Laokulrat, Makoto Miwa, Yoshimasa Tsuruoka and Takashi Chikayama. 2013. UTTime: Temporal relation classification using deep syntactic features. In *SemEval 2013*, pages 89–92.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee and James Pustejovsky. 2006. Machine Learning of Temporal Relations. In *ACL 2006*, pages 753–760.

Stephanie A. Miller and Lenhart K. Schubert. 1999. Time revisited. In *Computational Intelligence 6*, pages 108–118.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. In *Computational Linguistics*. 34(1). pages 35–80, MIT Press.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andew See, Rob Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro and Marcia Lazo. 2003. The TIMEBANK Corpus. In *Proceedings of Corpus Linguistics 2003 (March 2003)*, pages 545–557.

James Pustejovsky, Robert Ingria, Roser Saurí, José Castaño, Jessica Littman, Rob Gaizauskas, Andrea Setzer, Graham Katz and Inderjeet Mani. 2005. The specification language TimeML. In *The Language of Time: A reader*, pages 545–557.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a question answering system. In *ACL 2002*, pages 41–47.

Stanford Natural Language Processing Group. 2012. Stanford CoreNLP.

Marta Tatu and Munirathnam Srikanth. 2008. Experiments with reasoning for temporal relations between events. In *COLING 2008*, pages 857–864.

Naushad UzZaman and James F. Allen. 2011. Temporal evaluation. In *ACL 2011*, pages 351–356.

Naushad UzZaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen and James Pustejovsky. 2013. SemEval-2013 Task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *SemEval 2013*, pages 2–9.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *SemEval 2007*, pages 75–80.

David H. Wolpert. 1992. Stacked generalization. In *Neural Networks*, volume 5, pages 241–259.

Katsumasa Yoshikawa, Sebastian Riedel ,Masayuki Asahara and Yuji Matsumoto. 2009. Jointly identifying temporal relations with Markov Logic. In *ACL 2009*, pages 405–413.