NAIST.Japan: Temporal Relation Identification Using Dependency Parsed Tree

Yuchang Cheng, Masayuki Asahara and Yuji Matsumoto
Graduate School of Informatino Science,
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara, 630-0192, Japan
{yuchan-c, masayu-a, matsu}@is.naist.jp

Abstract

In this paper, we attempt to use a sequence labeling model with features from dependency parsed tree for temporal relation identification. In the sequence labeling model, the relations of contextual pairs can be used as features for relation identification of the current pair. Head-modifier relations between pairs of words within one sentence can be also used as the features. In our preliminary experiments, these features are effective for the temporal relation identification tasks.

1 Overview of our system

This paper presents a temporal relation identifier by the team NAIST.Japan. Our identifier has two characteristics: sequence labeling model and use of dependency parsed tree.

Firstly, we treated each problem a sequence labeling problem, such that event/time pairs were ordered by the position of the events and times in the document. This idea is for task B and C. In task B, the neighbouring relations between an EVENT and DCT-TIMEX3 tend to interact. In task C, when EVENT-a, EVENT-b, and EVENT-c are linearly ordered, the relation between EVENT-a and EVENT-b tends to affect the one between EVENT-b and EVENT-c.

Secondly, we introduced dependency features where each word was annotated with a label indicating its tree position to the event and the time, e.g. "descendant" of the event and "ancestor" of the time. The dependency features are introduced for our machine learning-based relation identifier. In task A, we need to label several different event-time pairs within the same sentence. We can use information from TIMEX3, which is a descendent of the target EVENT in the dependency tree.

Section 2 shows how to use a sequence labeling model for the task. Section 3 shows how to use the dependency parsed tree for the model. Section 4 presents the results and discussions.

2 Temporal Relation Identification by Sequence Labeling

Our approach to identify temporal relation is based on a sequence labeling model. The target pairs are linearly ordered in the texts.

Sequence labeling model can be defined as a method to estimate an optimal label sequence $y = \langle y_1, y_2, \ldots, y_n \rangle$ over an observed sequence $x = \langle x_1, x_2, \ldots, x_n \rangle$. We consider, w-parameterized function

$$f(x) = arg\max_{y \in \mathcal{Y}} F(x, y; w) = arg\max_{y \in \mathcal{Y}} \langle w, \Phi(x, y) \rangle.$$

Here, $\mathcal{Y}$ denotes all possible label combinations over y; $\Phi(x, y)$ denotes a feature expression over x, y. Introducing a kernel function:

$$K((x, y), (\bar{x}, \bar{y})) = \langle \Phi(x, y), \Phi(\bar{x}, \bar{y}) \rangle,$$

we have a dual representation:

$$F(x, y) = \sum_{i=1}^{m} \alpha_i K((\tilde{x}^{(i)}, \tilde{y}^{(i)}), (x, y)),$$

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 245–248,
Prague, June 2007. ©2007 Association for Computational Linguistics

given a training data set $\{(\tilde{x}^{(1)}, \tilde{y}^{(1)}), \ldots, (\tilde{x}^{(m)}, \tilde{y}^{(m)})\}$. We use HMM_SVM (Altun et al., 2003) as the sequence labeling model, in which the training is performed to maximize a margin

$$\gamma_j = F(\tilde{x}^{(k)}, \tilde{y}^{(k)}) - \max_{y \neq \tilde{y}^{(k)}} F(\tilde{x}^{(k)}, y).$$

The sequence labeling approach is natural for task B and C. In task B, if a document is about affairs in the past, the relations between events and a document creation time tend to be "BEFORE". All relations in task B depend on each other. In task C, if a relation between the preceding event and the current one is "AFTER", the current one is in the past. The information helps to determine the relation between the current and succeeding one. Whereas we have reasonable explanation to introduce sequence labeling for task B and C, we cannot for task A. However, in our preliminary experiments with trial data, the sequence labeling model outperformed point-wise models for task A. Thus, we introduce the sequence labeling model for task A.

Now, we present the sequence labeling approach for each task in detail by figure 1, 2 and 3. The left parts of figures are the graphical models of the sequence labeling. The right parts are the tagged corpus: ⟨S⟩ and ⟨/S⟩ are sentence boundaries; a EVENT-*nn* denotes an EVENT; a TIME-*nn* denotes a TIMEX3; a TIME-DCT in figure 2 denotes a TIMEX3 with document creation time; a boxed EVENT-*nn* in figure 3 denotes a matrix verb EVENT.

For task A (figure 1), x is a sequence of pairs between an EVENT and a TIMEX3 within the same sentence. y is a sequence of corresponding relations. Event-time pairs are ordered first by sentence position, then by event position and finally by time position. For task B (figure 2), x is a sequence of pairs between an EVENT and a DCT-TIMEX3. y is a sequence of corresponding relations. All pairs in the same text are linearly ordered and connected. For task C (figure 3), x is a sequence of pairs between two matrix verb EVENTs in the neighboring sentences. y is a sequence of corresponding relations. All pairs in the same text are linearly ordered and connected, even if the two relations are not in the adjacent sentences.

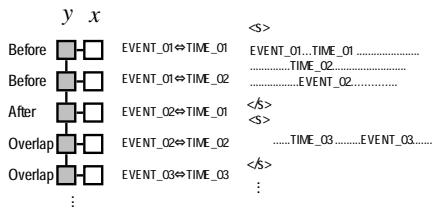


Figure 1: Sequence Labeling Model for Task A

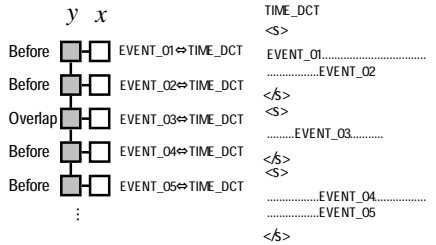


Figure 2: Sequence Labeling Model for Task B

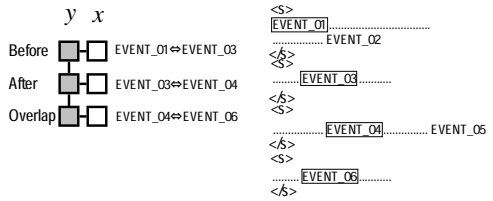


Figure 3: Sequence Labeling Model for Task C

3 Features from Dependency Parsed Tree

A dependency relation is a head-modifier relation on a syntactic tree. Figure 4 shows an example dependency parsed tree of the following sentence – "*The warrants may be exercised until 90 days after their issue date*". We parsed the TimeEval data using MSTParser v0.2 (McDonald and Pereira, 2006), which is trained with all Penn Treebank (Marcus et al., 1993) without dependency label.

We introduce **tree position** labels between an target node and another node on the dependency parsed tree: ANC (ancestor), DES (descendant), SIB (sibling), and TARGET (target word). Figure 5 shows the labels, in which the box with double lines is the target node. The tree position between the target EVENT and a word in the target TIMEX3 is used as a feature for our machine learning-based relation identifier.

We also use the words in the sentence including the target entities as features. Each word is anno-

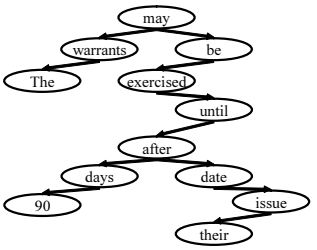


Figure 4: An example of dependency parsed tree

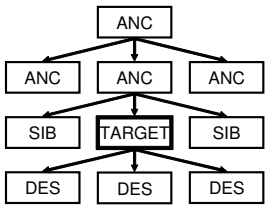
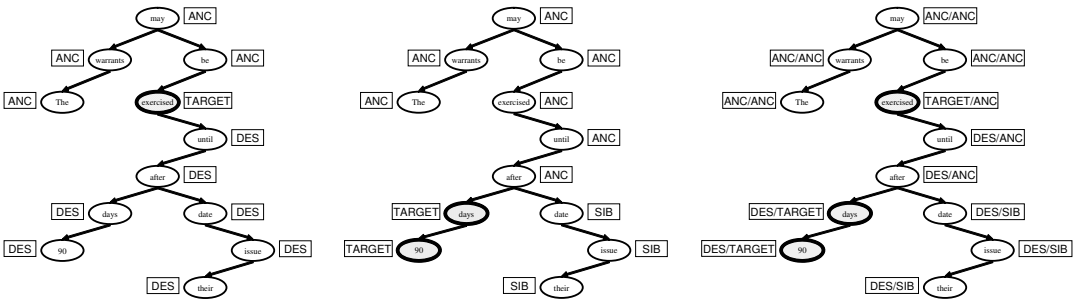


Figure 5: Tree position labels



TARGET node: "*exercised*" TARGET nodes: "*90*" and "*days*" TARGET-A node: "*exercised*"
TARGET-B nodes: "*90*" and "*days*"

(1) EVENT-based (2) TIMEX3-based (3) JOINT

Figure 6: Tree position labels on the example dependency parsed tree

tated with (1) its tree position to the EVENT, (2) its tree position to the TIMEX3, and (3) the combination of the labels from (1) and (2). Fig. 6 shows the labels of tree positions. The left picture shows (1) EVENT-based labels of the tree position with the target EVENT "*exercised*". The center picture shows (2) TIMEX3-based ones with the target TIMEX3 "*90 days*". The right picture shows (3) JOINT ones which are combinations of the relation label with the EVENT and with the TIMEX3. We perform feature selection on the words in the current sentence according to the tree position labels. Note that, when MSTparser outputs more than one trees for a sentence, we introduce a meta-root node to bundle the ones in a tree.

4 Results and Discussions

We use HMM_SVM [1]as a sequence labeling model with features in Table 1, 2 and 3 for task A, B and C, respectively. The attributes *value* in TIMEX3 is encoded as the relation with DCT-TIMEX3: {BEFORE, OVERLAP, AFTER, VAGUE}. In task A, only words in the current sentence with JOINT relation labels "TARGET/$*$" or "ANC/$*$" or "*/DES"[2] were used. In task C, attributes in the TIMEX3 are annotated with the flag whether the TIMEX3 entity is the highest (namely the nearest to the root node) in the tree. Some adverbs and conjunctions in the succeeding sentence help to determine the adjacent two relations. Thus, we introduce all words in the succeeding sentence for Task A and B. These features are determined by our preliminary experiments with the trial data .

Table 4 is our results on the test data. Whereas, our system is average rank in task A and B, it is worst mark in task C. The features from dependency parsed trees are effective for task A and B. However, these are not for task C.

Now, we focus on what went wrong instead of what went right in our preliminary experiments in trial data. We tried point-wise methods with other

[1] `http://svmlight.joachims.org/svm_struct.html`

[2] '$*$' stands for wild cards.

Table 1: Features for Task A

all attributes in the target EVENT
all attributes in the target TIMEX3 −the attributes *value* is encoded as the relation with DCT-TIMEX3
all words in the current sentence with TIMEX3-based label (2) of tree position
words in the current sentence with JOINT label (3) of tree position − only relation label with "TARGET/∗" or "ANC/∗" or "∗/DES" (∗ stands for wild cards)
label (1) of tree position from the EVENT to the TIMEX3
all words in the succeeding sentence

Table 2: Features for Task B

all attributes in the target EVENT
all attributes in the target TIMEX3 of in the current sentence with EVENT-based label (1) of tree position
all attributes in the target TIMEX3 of in the preceding and succeeding sentence
all words in the current sentence with EVENT-based label (1) of tree position
all words in the succeeding sentence

Table 3: Features for Task C

all attributes in the target two EVENTs (EVENT-1 and EVENT-2)
all attributes in the TIMEX3 in the sentence including EVENT-1 with the label (1) of tree position to EVENT-1
all attributes in the TIMEX3 in the sentence including EVENT-2 with the label (1) of tree position to EVENT-2
all words in the sentence including EVENT-1 with the label (1) of tree position to EVENT-1
all words in the sentence including EVENT-2 with the label (1) of tree position to EVENT-2

machine learners such as maximum entropy and multi-class support vector machines. However, sequence labeling method with HMM_SVM outperformed other point-wise methods in the trial data.

We have dependency parsed trees of the sentences. Naturally, it would be effective to introduce point-wise tree-based classifiers such as Tree Kernels in SVM (Collins and Duffy, 2002; Vishwanathan and Smola, 2002) and boosting for classification of trees (Kudo and Matsumoto, 2004). We tried a boosting learner [3] which enables us to perform subtree feature selection for the tasks. However, the boosting learner selected only one-node subtrees as useful features. Thus, we perform simple vector-based feature engineering on HMM_SVM.

[3] http://chasen.org/~taku/software/bact/

Table 4: Results

Task	P	R	F	Rank
Task A (strict)	0.61	0.61	0.61	2/6
Task A (relaxed)	0.63	0.63	0.63	2/6
Task B (strict)	0.75	0.75	0.75	2/6
Task B (relaxed)	0.76	0.76	0.76	2/6
Task C (strict)	0.49	0.49	0.49	5/6
Task C (relaxed)	0.56	0.56	0.56	6/6

We believe that it is necessary for solving task C to incorporate knowledge of verb-verb relation. We also tried to use features in verb ontology such as VERBOCEAN (Chklovsky and Pantel, 2004) which is used in (Mani et al., 2006). It did not improved performance in our preliminary experiments with trial data.

References

Y. Altun, I. Tsochantaridis, and T. Hofmann. 2003. Hidden markov support vector machines. In *Proc. of ICML-2003*.

T. Chklovsky and P. Pantel. 2004. Verbocean: Mining the web for fine-grained semantiv verb relations. In *Proc. of EMNLP-2004*.

M. Collins and N. Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proc. of ACL-2002*.

T. Kudo and Y. Matsumoto. 2004. A boosting algorithm for classification of semi-structured text. In *Proc. of EMNLP-2004*.

I. Mani, M. Verhagen, B. Wellner, C. M. Lee, and J. Pustejovsky. 2006. Machine learning of temporal relations. In *Proc. of ACL-2006*.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. 19(2):313–330.

R. McDonald and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proc. of EACL-2006*.

M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, and J. Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proc. of SemEval-2007*.

S. V. N. Vishwanathan and A. J. Smola. 2002. Fast kernels on strings and trees. In *Proc. of NIPS-2002*.