Jointly Identifying Temporal Relations with Markov Logic

Katsumasa Yoshikawa NAIST, Japan katsumasa-y@is.naist.jp
Sebastian Riedel University of Tokyo, Japan sebastian.riedel@gmail.com
Masayuki Asahara NAIST, Japan masayu-a@is.naist.jp
Yuji Matsumoto NAIST, Japan matsu@is.naist.jp

Abstract

Recent work on temporal relation identification has focused on three types of relations between events: temporal relations between an event and a time expression, between a pair of events and between an event and the document creation time. These types of relations have mostly been identified in isolation by event pairwise comparison. However, this approach neglects logical constraints between temporal relations of different types that we believe to be helpful. We therefore propose a Markov Logic model that jointly identifies relations of all three relation types simultaneously. By evaluating our model on the TempEval data we show that this approach leads to about 2% higher accuracy for all three types of relations —and to the best results for the task when compared to those of other machine learning based systems.

1 Introduction

Temporal relation identification (or temporal ordering) involves the prediction of temporal order between events and/or time expressions mentioned in text, as well as the relation between events in a document and the time at which the document was created.

With the introduction of the TimeBank corpus (Pustejovsky et al., 2003), a set of documents annotated with temporal information, it became possible to apply machine learning to temporal ordering (Boguraev and Ando, 2005; Mani et al., 2006). These tasks have been regarded as essential for complete document understanding and are useful for a wide range of NLP applications such as question answering and machine translation.

Most of these approaches follow a simple schema: they learn classifiers that predict the temporal order of a given event pair based on a set of the pair's of features. This approach is *local* in the sense that only a single temporal relation is considered at a time.

Learning to predict temporal relations in this isolated manner has at least two advantages over any approach that considers several temporal relations jointly. First, it allows us to use off-the-shelf machine learning software that, up until now, has been mostly focused on the case of local classifiers. Second, it is computationally very efficient both in terms of training and testing.

However, the local approach has a inherent drawback: it can lead to solutions that violate logical constraints we know to hold for any sets of temporal relations. For example, by classifying temporal relations in isolation we may predict that event A happened before, and event B after, the time of document creation, but also that event A happened after event B—a clear contradiction in terms of temporal logic.

In order to repair the contradictions that the local classifier predicts, Chambers and Jurafsky (2008) proposed a global framework based on Integer Linear Programming (ILP). They showed that large improvements can be achieved by explicitly incorporating temporal constraints.

The approach we propose in this paper is similar in spirit to that of Chambers and Jurafsky: we seek to improve the accuracy of temporal relation identification by predicting relations in a more global manner. However, while they focused only on the temporal relations between events mentioned in a document, we also jointly predict the temporal order between events and time expressions, and between events and the document creation time.

Our work also differs in another important aspect from the approach of Chambers and Jurafsky. Instead of combining the output of a set of local classifiers using ILP, we approach the problem of joint temporal relation identification using Markov Logic (Richardson and Domingos, 2006). In this

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 405–413,
Suntec, Singapore, 2-7 August 2009.

framework global correlations can be readily captured through the addition of weighted first order logic formulae.

Using Markov Logic instead of an ILP-based approach has at least two advantages. First, it allows us to easily capture non-deterministic (soft) rules that tend to hold between temporal relations but do not have to. [1] For example, if event A happens before B, and B overlaps with C, then there is a good chance that A also happens before C, but this is not guaranteed.

Second, the amount of engineering required to build our system is similar to the efforts required for using an off-the-shelf classifier: we only need to define features (in terms of formulae) and provide input data in the correct format. [2] In particular, we do not need to manually construct ILPs for each document we encounter. Moreover, we can exploit and compare advanced methods of global inference and learning, as long as they are implemented in our Markov Logic interpreter of choice. Hence, in our future work we can focus entirely on temporal relations, as opposed to inference or learning techniques for machine learning.

We evaluate our approach using the data of the "TempEval" challenge held at the SemEval 2007 Workshop (Verhagen et al., 2007). This challenge involved three tasks corresponding to three types of temporal relations: between events and time expressions in a sentence (Task A), between events of a document and the document creation time (Task B), and between events in two consecutive sentences (Task C).

Our findings show that by incorporating global constraints that hold between temporal relations predicted in Tasks A, B and C, the accuracy for all three tasks can be improved significantly. In comparison to other participants of the "TempEval" challenge our approach is very competitive: for two out of the three tasks we achieve the best results reported so far, by a margin of at least 2%. [3] Only for Task B we were unable to reach the performance of a rule-based entry to the challenge. However, we do perform better than all pure machine learning-based entries.

The remainder of this paper is organized as follows: Section 2 describes temporal relation identification including TempEval; Section 3 introduces Markov Logic; Section 4 explains our proposed Markov Logic Network; Section 5 presents the setup of our experiments; Section 6 shows and discusses the results of our experiments; and in Section 7 we conclude and present ideas for future research.

2 Temporal Relation Identification

Temporal relation identification aims to predict the temporal order of events and/or time expressions in documents, as well as their relations to the document creation time (DCT). For example, consider the following (slightly simplified) sentence of Section 1 in this paper.

> With the introduction of the TimeBank corpus (Pustejovsky et al., 2003), machine learning approaches to temporal ordering became possible.

Here we have to predict that the "Machine learning becoming possible" event happened *AFTER* the "introduction of the TimeBank corpus" event, and that it has a temporal *OVERLAP* with the year 2003. Moreover, we need to determine that both events happened *BEFORE* the time this paper was created.

Most previous work on temporal relation identification (Boguraev and Ando, 2005; Mani et al., 2006; Chambers and Jurafsky, 2008) is based on the TimeBank corpus. The temporal relations in the Timebank corpus are divided into 11 classes; 10 of them are defined by the following 5 relations and their inverse: *BEFORE*, *IBEFORE (immediately before)*, *BEGINS*, *ENDS*, *INCLUDES*; the remaining one is *SIMULTANEOUS*.

In order to drive forward research on temporal relation identification, the SemEval 2007 shared task (Verhagen et al., 2007) (TempEval) included the following three tasks.

TASK A Temporal relations between events and time expressions that occur within the same sentence.

TASK B Temporal relations between the Document Creation Time (DCT) and events.

TASK C Temporal relations between the main events of adjacent sentences.[4]

[1] It is clearly possible to incorporate weighted constraints into ILPs, but how to learn the corresponding weights is not obvious.

[2] This is not to say that picking the right formulae in Markov Logic, or features for local classification, is always easy.

[3] To be slightly more precise: for Task C we achieve this margin only for "strict" scoring—see sections 5 and 6 for more details.

[4] The main event of a sentence is expressed by its syntactically dominant verb.

To simplify matters, in the TempEval data, the classes of temporal relations were reduced from the original 11 to 6: *BEFORE, OVERLAP, AFTER, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER*, and *VAGUE*.

In this work we are focusing on the three tasks of TempEval, and our running hypothesis is that they should be tackled *jointly*. That is, instead of learning separate probabilistic models for each task, we want to learn a single one for all three tasks. This allows us to incorporate rules of temporal consistency that should hold across tasks. For example, if an event X happens *before* DCT, and another event Y *after* DCT, then surely X should have happened *before* Y. We illustrate this type of transition rule in Figure 1.

Note that the correct temporal ordering of events and time expressions can be controversial. For instance, consider the example sentence again. Here one could argue that "the introduction of the TimeBank" may *OVERLAP* with "Machine learning becoming possible" because "introduction" can be understood as a process that is not finished with the release of the data but also includes later advertisements and announcements. This is reflected by the low inter-annotator agreement score of 72% on Tasks A and B, and 68% on Task C.

3 Markov Logic

It has long been clear that local classification alone cannot adequately solve all prediction problems we encounter in practice.[5] This observation motivated a field within machine learning, often referred to as Statistical Relational Learning (SRL), which focuses on the incorporation of global correlations that hold between statistical variables (Getoor and Taskar, 2007).

One particular SRL framework that has recently gained momentum as a platform for global learning and inference in AI is Markov Logic (Richardson and Domingos, 2006), a combination of first-order logic and Markov Networks. It can be understood as a formalism that extends first-order logic to allow formulae that can be violated with some penalty. From an alternative point of view, it is an expressive template language that uses first order logic formulae to instantiate Markov Networks of repetitive structure.

From a wide range of SRL languages we chose Markov Logic because it supports discriminative

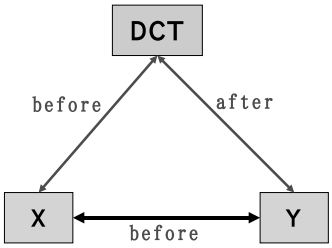

Figure 1: Example of Transition Rule 1

training (as opposed to generative SRL languages such as PRM (Koller, 1999)). Moreover, several Markov Logic software libraries exist and are freely available (as opposed to other discriminative frameworks such as Relational Markov Networks (Taskar et al., 2002)).

In the following we will explain Markov Logic by example. One usually starts out with a set of predicates that model the decisions we need to make. For simplicity, let us assume that we only predict two types of decisions: whether an event e happens before the document creation time (DCT), and whether, for a pair of events e_1 and e_2, e_1 happens before e_2. Here the first type of decision can be modeled through a unary predicate beforeDCT(e), while the latter type can be represented by a binary predicate before(e_1, e_2). Both predicates will be referred to as *hidden* because we do not know their extensions at test time. We also introduce a set of *observed* predicates, representing information that is available at test time. For example, in our case we could introduce a predicate futureTense(e) which indicates that e is an event described in the future tense.

With our predicates defined, we can now go on to incorporate our intuition about the task using weighted first-order logic formulae. For example, it seems reasonable to assume that

$$\text{futureTense}\,(e) \Rightarrow \neg\text{beforeDCT}\,(e) \qquad (1)$$

often, but not always, holds. Our remaining uncertainty with regard to this formula is captured by a weight w we associate with it. Generally we can say that the larger this weight is, the more likely/often the formula holds in the solutions described by our model. Note, however, that we do not need to manually pick these weights; instead they are learned from the given training corpus.

The intuition behind the previous formula can also be captured using a local classifier.[6] However,

[5] It can, however, solve a large number of problems surprisingly well.

[6] Consider a log-linear binary classifier with a "past-tense"

Markov Logic also allows us to say more:

$$\text{beforeDCT}(e_1) \wedge \neg\text{beforeDCT}(e_2) \Rightarrow \text{before}(e_1, e_2) \quad (2)$$

In this case, we made a statement about more global properties of a temporal ordering that cannot be captured with local classifiers. This formula is also an example of the transition rules as seen in Figure 2. This type of rule forms the core idea of our joint approach.

A *Markov Logic Network* (MLN) M is a set of pairs (ϕ, w) where ϕ is a first order formula and w is a real number (the formula's weight). It defines a probability distribution over sets of ground atoms, or so-called *possible worlds*, as follows:

$$p(\mathbf{y}) = \frac{1}{Z} \exp\left(\sum_{(\phi,w)\in M} w \sum_{\mathbf{c}\in C^\phi} f_{\mathbf{c}}^{\phi}(\mathbf{y})\right) \quad (3)$$

Here each $\mathbf{c}$ is a binding of free variables in ϕ to constants in our domain. Each $f_{\mathbf{c}}^{\phi}$ is a binary feature function that returns 1 if in the possible world $\mathbf{y}$ the *ground formula* we get by replacing the free variables in ϕ with the constants in $\mathbf{c}$ is true, and 0 otherwise. C^ϕ is the set of all bindings for the free variables in ϕ. Z is a normalisation constant. Note that this distribution corresponds to a Markov Network (the so-called *Ground Markov Network*) where nodes represent ground atoms and factors represent ground formulae.

Designing formulae is only one part of the game. In practice, we also need to choose a training regime (in order to learn the weights of the formulae we added to the MLN) and a search/inference method that picks the most likely set of ground atoms (temporal relations in our case) given our trained MLN and a set of observations. However, implementations of these methods are often already provided in existing Markov Logic interpreters such as *Alchemy* [7] and *Markov thebeast*. [8]

4 Proposed Markov Logic Network

As stated before, our aim is to jointly tackle Tasks A, B and C of the TempEval challenge. In this section we introduce the Markov Logic Network we designed for this goal.

We have three hidden predicates, corresponding to Tasks A, B, and C: relE2T(e, t, r) represents the temporal relation of class r between an event e and a time expression t; relDCT(e, r) denotes the temporal relation r between an event e and DCT; relE2E$(e1, e2, r)$ represents the relation r between two events of the adjacent sentences, $e1$ and $e2$.

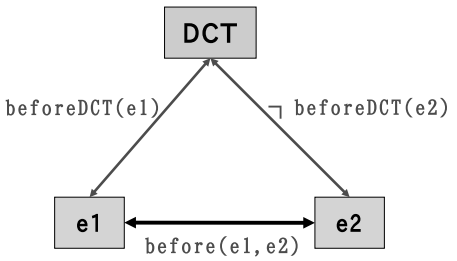

Figure 2: Example of Transition Rule 2

Our observed predicates reflect information we were given (such as the words of a sentence), and additional information we extracted from the corpus (such as POS tags and parse trees). Note that the TempEval data also contained temporal relations that were not supposed to be predicted. These relations are represented using two observed predicates: relT2T$(t1, t2, r)$ for the relation r between two time expressions $t1$ and $t2$; dctOrder(t, r) for the relation r between a time expression t and a fixed DCT.

An illustration of all "temporal" predicates, both hidden and observed, can be seen in Figure 3.

4.1 Local Formula

Our MLN is composed of several weighted formulae that we divide into two classes. The first class contains *local* formulae for the Tasks A, B and C. We say that a formula is local if it only considers the hidden temporal relation of a single event-event, event-time or event-DCT pair. The formulae in the second class are *global*: they involve two or more temporal relations at the same time, and consider Tasks A, B and C simultaneously.

The local formulae are based on features employed in previous work (Cheng et al., 2007; Bethard and Martin, 2007) and are listed in Table 1. What follows is a simple example in order to illustrate how we implement each feature as a formula (or set of formulae).

Consider the tense-feature for Task C. For this feature we first introduce a predicate tense(e, t) that denotes the tense t for an event e. Then we

feature: here for every event e the decision "e happens before DCT" becomes more likely with a higher weight for this feature.

[7] http://alchemy.cs.washington.edu/

[8] http://code.google.com/p/thebeast/

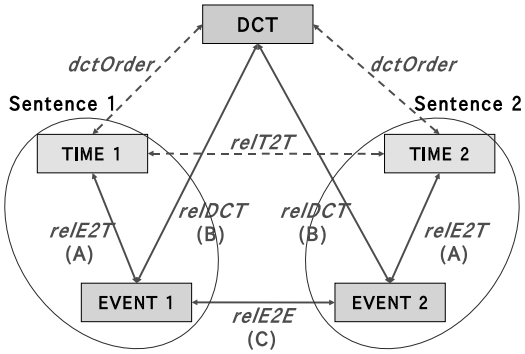

Figure 3: Predicates for Joint Formulae; observed predicates are indicated with dashed lines.

Table 1: Local Features

Feature	A	B	C
EVENT-word	X		X
EVENT-POS	X		X
EVENT-stem	X		X
EVENT-aspect	X	X	X
EVENT-tense	X	X	X
EVENT-class	X	X	X
EVENT-polarity	X		X
TIMEX3-word	X		
TIMEX3-POS	X		
TIMEX3-value	X		
TIMEX3-type	X		
TIMEX3-DCT order	X	X	
positional order	X		
in/outside	X		
unigram(word)	X		X
unigram(POS)	X		X
bigram(POS)	X		
trigram(POS)	X		X
Dependency-Word	X	X	X
Dependency-POS	X	X	

add a set of formulae such as

$$\text{tense}(e1, \text{past}) \wedge \text{tense}(e2, \text{future}) \Rightarrow \text{relE2E}(e1, e2, \text{before}) \quad (4)$$

for all possible combinations of tenses and temporal relations.[9]

4.2 Global Formula

Our global formulae are designed to enforce consistency between the three hidden predicates (and the two observed temporal predicates we mentioned earlier). They are based on the transition rules we mentioned in Section 3.

Table 2 shows the set of formula templates we use to generate the global formulae. Here each template produces several instantiations, one for each assignment of temporal relation classes to the variables R1, R2, etc. One example of a template instantiation is the following formula.

$$\text{dctOrder}(t1, \text{before}) \wedge \text{relDCT}(e1, \text{after}) \Rightarrow \text{relE2T}(e1, t1, \text{after}) \quad (5a)$$

This formula is an expansion of the formula template in the second row of Table 2. Note that it utilizes the results of Task B to solve Task A.

Formula 5a should always hold,[10] and hence we could easily implement it as a hard constraint in an ILP-based framework. However, some transition rules are less determinstic and should rather be taken as "rules of thumb". For example, formula 5b is a rule which we expect to hold often, but not always.

$$\text{dctOrder}(t1, \text{before}) \wedge \text{relDCT}(e1, \text{overlap}) \Rightarrow \text{relE2T}(e1, t1, \text{after}) \quad (5b)$$

Fortunately, this type of soft rule poses no problem for Markov Logic: after training, Formula 5b will simply have a lower weight than Formula 5a. By contrast, in a "Local Classifier + ILP"-based approach as followed by Chambers and Jurafsky (2008) it is less clear how to proceed in the case of soft rules. Surely it is possible to incorporate weighted constraints into ILPs, but how to learn the corresponding weights is not obvious.

5 Experimental Setup

With our experiments we want to answer two questions: (1) does jointly tackling Tasks A, B, and C help to increase overall accuracy of temporal relation identification? (2) How does our approach compare to state-of-the-art results? In the following we will present the experimental set-up we chose to answer these questions.

In our experiments we use the test and training sets provided by the TempEval shared task. We further split the original training data into a training and a development set, used for optimizing parameters and formulae. For brevity we will refer to the training, development and test set as TRAIN, DEV and TEST, respectively. The numbers of temporal relations in TRAIN, DEV, and TEST are summarized in Table 3.

[9] This type of "template-based" formulae generation can be performed automatically by the Markov Logic Engine.

[10] However, due to inconsistent annotations one will find violations of this rule in the TempEval data.

Table 2: Joint Formulae for Global Model

Task	Formula
$A \rightarrow B$	$\text{dctOrder}(t, R1) \wedge \text{relE2T}(e, t, R2) \Rightarrow \text{relDCT}(e, R3)$
$B \rightarrow A$	$\text{dctOrder}(t, R1) \wedge \text{relDCT}(e, R2) \Rightarrow \text{relE2T}(e, t, R3)$
$B \rightarrow C$	$\text{relDCT}(e1, R1) \wedge \text{relDCT}(e2, R2) \Rightarrow \text{relE2E}(e1, e2, R3)$
$C \rightarrow B$	$\text{relDCT}(e1, R1) \wedge \text{relE2E}(e1, e2, R2) \Rightarrow \text{relDCT}(e2, R3)$
$A \rightarrow C$	$\text{relE2T}(e1, t1, R1) \wedge \text{relT2T}(t1, t2, R2) \wedge \text{relE2T}(e2, t2, R3) \Rightarrow \text{relE2E}(e1, e2, R4)$
$C \rightarrow A$	$\text{relE2T}(e2, t2, R1) \wedge \text{relT2T}(t1, t2, R2) \wedge \text{relE2E}(e1, e2, R3) \Rightarrow \text{relE2T}(e1, t1, R4)$

Table 3: Numbers of Labeled Relations for All Tasks

	TRAIN	DEV	TEST	TOTAL
Task A	1359	131	169	1659
Task B	2330	227	331	2888
Task C	1597	147	258	2002

For feature generation we use the following tools. [11] POS tagging is performed with TnT ver2.2;[12] for our dependency-based features we use MaltParser 1.0.0.[13] For inference in our models we use Cutting Plane Inference (Riedel, 2008) with ILP as a base solver. This type of inference is exact and often very fast because it avoids instantiation of the complete Markov Network. For learning we apply one-best MIRA (Crammer and Singer, 2003) with Cutting Plane Inference to find the current model guess. Both training and inference algorithms are provided by *Markov thebeast*, a Markov Logic interpreter tailored for NLP applications.

Note that there are several ways to manually optimize the set of formulae to use. One way is to pick a task and then choose formulae that increase the accuracy for this task on DEV. However, our primary goal is to improve the performance of all the tasks together. Hence we choose formulae with respect to the total score over all three tasks. We will refer to this type of optimization as "averaged optimization". The total scores of the all three tasks are defined as follows:

$$\frac{C_a + C_b + C_c}{G_a + G_b + G_c}$$

where C_a, C_b, and C_c are the number of the correctly identified labels in each task, and G_a, G_b, and G_c are the numbers of gold labels of each task. Our system necessarily outputs one label to one relational link to identify. Therefore, for all our results, precision, recall, and F-measure are the exact same value.

For evaluation, TempEval proposed the two scoring schemes: "strict" and "relaxed". For strict scoring we give full credit if the relations match, and no credit if they do not match. On the other hand, relaxed scoring gives credit for a relation according to Table 4. For example, if a system picks the relation "AFTER" that should have been "BEFORE" according to the gold label, it gets neither "strict" nor "relaxed" credit. But if the system assigns "B-O (BEFORE-OR-OVERLAP)" to the relation, it gets a 0.5 "relaxed" score (and still no "strict" score).

6 Results

In the following we will first present our comparison of the local and global model. We will then go on to put our results into context and compare them to the state-of-the-art.

6.1 Impact of Global Formulae

First, let us show the results on TEST in Table 5. You will find two columns, "Global" and "Local", showing scores achieved with and without joint formulae, respectively. Clearly, the global models scores are higher than the local scores for all three tasks. This is also reflected by the last row of Table 5. Here we see that we have improved the averaged performance across the three tasks by approximately 2.5% ($\rho < 0.01$, McNemar's test 2-tailed). Note that with 3.5% the improvements are particularly large for Task C.

The TempEval test set is relatively small (see Table 3). Hence it is not clear how well our results would generalize in practice. To overcome this issue, we also evaluated the local and global model using 10-fold cross validation on the training data (TRAIN + DEV). The corresponding results can be seen in Table 6. Note that the general picture remains: performance for all tasks is improved, and the averaged score is improved only slightly less than for the TEST results. However, this time the score increase for Task B is lower than before. We

[11] Since the TempEval trial has no restriction on preprocessing such as syntactic parsing, most participants used some sort of parsers.

[12] http://www.coli.uni-saarland.de/~thorsten/tnt/

[13] http://w3.msi.vxu.se/~nivre/research/MaltParser.html

Table 4: Evaluation Weights for Relaxed Scoring

	B	O	A	B-O	O-A	V
B	1	0	0	0.5	0	0.33
O	0	1	0	0.5	0.5	0.33
A	0	0	1	0	0.5	0.33
B-O	0.5	0.5	0	1	0.5	0.67
O-A	0	0.5	0.5	0.5	1	0.67
V	0.33	0.33	0.33	0.67	0.67	1

B: BEFORE O: OVERLAP
A: AFTER B-O: BEFORE-OR-OVERLAP
O-A: OVERLAP-OR-AFTER V: VAGUE

Table 5: Results on TEST Set

	Local		Global	
task	strict	relaxed	strict	relaxed
Task A	0.621	0.669	0.645	0.687
Task B	0.737	0.753	0.758	0.777
Task C	0.531	0.599	0.566	0.632
All	0.641	0.682	0.668	0.708

Table 6: Results with 10-fold Cross Validation

	Local		Global	
task	strict	relaxed	strict	relaxed
Task A	0.613	0.645	0.662	0.691
Task B	0.789	0.810	0.799	0.819
Task C	0.533	0.608	0.552	0.623
All	0.667	0.707	0.689	0.727

see that this is compensated by much higher scores for Task A and C. Again, the improvements for all three tasks are statistically significant ($\rho < 10^{-8}$, McNemar's test, 2-tailed).

To summarize, we have shown that by tightly connecting tasks A, B and C, we can improve temporal relation identification significantly. But are we just improving a weak baseline, or can joint modelling help to reach or improve the state-of-the-art results? We will try to answer this question in the next section.

6.2 Comparison to the State-of-the-art

In order to put our results into context, Table 7 shows them along those of other TempEval participants. In the first row, TempEval Best gives the best scores of TempEval for each task. Note that all but the strict scores of Task C are achieved by WVALI (Puscasu, 2007), a hybrid system that combines machine learning and hand-coded rules. In the second row we see the TempEval average scores of all six participants in TempEval. The third row shows the results of CU-TMP (Bethard and Martin, 2007), an SVM-based system that achieved the second highest scores in TempEval for all three tasks. CU-TMP is of interest because it is the best pure Machine-Learning-based approach so far.

The scores of our local and global model come in the fourth and fifth row, respectively. The last row in the table shows task-adjusted scores. Here we essentially designed and applied three global MLNs, each one tailored and optimized for a different task. Note that the task-adjusted scores are always about 1% higher than those of the single global model.

Let us discuss the results of Table 7 in detail. We see that for task A, our global model improves an already strong local model to reach the best results both for strict scores (with a 3% points margin) and relaxed scores (with a 5% points margin).

For Task C we see a similar picture: here adding global constraints helped to reach the best strict scores, again by a wide margin. We also achieve competitive relaxed scores which are in close range to the TempEval best results.

Only for task B our results cannot reach the best TempEval scores. While we perform slightly better than the second-best system (CU-TMP), and hence report the best scores among all pure Machine-Learning based approaches, we cannot quite compete with WVALI.

6.3 Discussion

Let us discuss some further characteristics and advantages of our approach. First, notice that global formulae not only improve strict but also relaxed scores for all tasks. This suggests that we produce more ambiguous labels (such as BEFORE-OR-OVERLAP) in cases where the local model has been overconfident (and wrongly chose BEFORE or OVERLAP), and hence make less "fatal errors". Intuitively this makes sense: global consistency is easier to achieve if our labels remain ambiguous. For example, a solution that labels every relation as VAGUE is globally consistent (but not very informative).

Secondly, one could argue that our solution to joint temporal relation identification is too complicated. Instead of performing global inference, one could simply arrange local classifiers for the tasks into a pipeline. In fact, this has been done by Bethard and Martin (2007): they first solve task B and then use this information as features for Tasks A and C. While they do report improvements (0.7%

Table 7: Comparison with Other Systems

	Task A		Task B		Task C	
	strict	relaxed	strict	relaxed	strict	relaxed
TempEval Best	0.62	0.64	**0.80**	**0.81**	0.55	**0.64**
TempEval Average	0.56	0.59	0.74	0.75	0.51	0.58
CU-TMP	0.61	0.63	0.75	0.76	0.54	0.58
Local Model	0.62	0.67	0.74	0.75	0.53	0.60
Global Model	**0.65**	**0.69**	0.76	0.78	**0.57**	0.63
Global Model (Task-Adjusted)	(0.66)	(0.70)	(0.76)	(0.79)	(0.58)	(0.64)

on Task A, and about 0.5% on Task C), generally these improvements do not seem as significant as ours. What is more, by design their approach can not improve the first stage (Task B) of the pipeline.

On the same note, we also argue that our approach does not require more implementation efforts than a pipeline. Essentially we only have to provide features (in the form of formulae) to the Markov Logic Engine, just as we have to provide for a SVM or MaxEnt classifier.

Finally, it became more clear to us that there are problems inherent to this task and dataset that we cannot (or only partially) solve using global methods. First, there are inconsistencies in the training data (as reflected by the low inter-annotator agreement) that often mislead the learner—this problem applies to learning of local and global formulae/features alike. Second, the training data is relatively small. Obviously, this makes learning of reliable parameters more difficult, particularly when data is as noisy as in our case. Third, the temporal relations in the TempEval dataset only directly connect a small subset of events. This makes global formulae less effective.[14]

7 Conclusion

In this paper we presented a novel approach to temporal relation identification. Instead of using local classifiers to predict temporal order in a pairwise fashion, our approach uses Markov Logic to incorporate both local features and global transition rules between temporal relations.

We have focused on transition rules between temporal relations of the three TempEval subtasks: temporal ordering of events, of events and time expressions, and of events and the document creation time. Our results have shown that global transition rules lead to significantly higher accuracy for all three tasks. Moreover, our global Markov Logic model achieves the highest scores reported so far for two of three tasks, and very competitive results for the remaining one.

While temporal transition rules can also be captured with an Integer Linear Programming approach (Chambers and Jurafsky, 2008), Markov Logic has at least two advantages. First, handling of "rules of thumb" between less specific temporal relations (such as OVERLAP or VAGUE) is straightforward—we simply let the Markov Logic Engine learn weights for these rules. Second, there is less engineering overhead for us to perform, because we do not need to generate ILPs for each document.

However, potential for further improvements through global approaches seems to be limited by the sparseness and inconsistency of the data. To overcome this problem, we are planning to use external or untagged data along with methods for unsupervised learning in Markov Logic (Poon and Domingos, 2008).

Furthermore, TempEval-2 [15] is planned for 2010 and it has challenging temporal ordering tasks in five languages. So, we would like to investigate the utility of global formulae for multilingual temporal ordering. Here we expect that while lexical and syntax-based features may be quite language dependent, global transition rules should hold across languages.

Acknowledgements

This work is partly supported by the Integrated Database Project, Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

Steven Bethard and James H. Martin. 2007. Cu-tmp: Temporal relation classification using syntactic and semantic features. In *Proceedings of the 4th International Workshop on SemEval-2007.*, pages 129–132.

[14] See (Chambers and Jurafsky, 2008) for a detailed discussion of this problem, and a possible solution for it.

[15] http://www.timeml.org/tempeval2/

Branimir Boguraev and Rie Kubota Ando. 2005. Timeml-compliant text analysis for temporal reasoning. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 997–1003.

Nathanael Chambers and Daniel Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 698–706, Honolulu, Hawaii, October. Association for Computational Linguistics.

Yuchang Cheng, Masayuki Asahara, and Yuji Matsumoto. 2007. Naist.japan: Temporal relation identification using dependency parsed tree. In *Proceedings of the 4th International Workshop on SemEval-2007.*, pages 245–248.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

Lise Getoor and Ben Taskar. 2007. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Daphne Koller, 1999. *Probabilistic Relational Models*, pages 3–13. Springer, Berlin/Heidelberg, Germany.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 753–760, Morristown, NJ, USA. Association for Computational Linguistics.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 650–659, Honolulu, Hawaii, October. Association for Computational Linguistics.

Georgiana Puscasu. 2007. Wvali: Temporal relation identification by syntactico-semantic analysis. In *Proceedings of the 4th International Workshop on SemEval-2007.*, pages 484–487.

James Pustejovsky, Jose Castano, Robert Ingria, Reser Sauri, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. The timebank corpus. In *Proceedings of Corpus Linguistics 2003*, pages 647–656.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. In *Machine Learning*.

Sebastian Riedel. 2008. Improving the accuracy and efficiency of map inference for markov logic. In *Proceedings of UAI 2008*.

Ben Taskar, Abbeel Pieter, and Daphne Koller. 2002. Discriminative probabilistic models for relational data. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 485–492, San Francisco, CA. Morgan Kaufmann.

Marc Verhagen, Robert Gaizaukas, 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 SemEval-2007.*, pages 75–80.