Long-Distance Time-Event Relation Extraction

Alessandro Moschitti
QCRI, Qatar Foundation, Doha, Qatar
DISI, University of Trento, Povo (TN), Italy
amoschitti@qf.org.qa
moschitti@disi.unitn.it

Siddharth Patwardhan and **Chris Welty**
IBM T. J. Watson Research Center
Yorktown Heights NY 10598
siddharth@us.ibm.com
welty@us.ibm.com

Abstract

This paper proposes state-of-the-art models for time-event relation extraction (TERE). The models are specifically designed to work effectively with relations that span multiple sentences and paragraphs, i.e., inter-sentence TERE. Our main idea is: (i) to build a computational representation of the context of the two target relation arguments, and (ii) to encode it as structural features in Support Vector Machines using tree kernels. Results on two data sets – Machine Reading and TimeBank – with 3-fold cross-validation show that the combination of traditional feature vectors and the new structural features improves on the state of the art for inter-sentence TERE by about 20%, achieving a 30.2 F1 score on inter-sentence TERE alone, and 47.2 F1 for all TERE (inter and intra sentence combined).

1 Introduction

Time-Event Relation Extraction (TERE) is the task of linking event mentions and relation mentions to occurrences of "time stamps" in text. We define it as follows: given a set of textual expressions denoting events and relations, and a set of time expressions in the same text document, find all instances of temporal relations between elements of the two input sets. A relation between an event and a time expression indicates that the event occurs within the temporal context specified by the time expression, for example, the following sentence:

He succeeded James A. Taylor, who stepped down as chairman, president and chief executive in March for health reasons; the appointment took effect Nov. 13.

conveys two different events, *succession* and *stepping down*, linked to the time stamps, *Nov. 13* and *March*, respectively.

In this paper, we focus on the task of linking time expressions to events, i.e., we carry out a classification task, where, for each possible pair of (event/relation, time) in a document, the classifier decides whether there exists a link between the two. In particular, we assume that the event mentions, relation mentions and time expressions are given to us by an external process. There is a large body of work on the above topics and they remain difficult problems, but we use human annotated mentions and expressions as input to our models since TERE itself is a relatively new problem in this context. Previous work in TempEval-2 (Verhagen et al., 2010) and our work (Hovy et al., 2012) have shown that accurate relation classifiers can be modeled with supervised approaches, provided that the expressions are limited to be in the same sentence. In contrast, there is almost no previous work on inter-sentence TERE (ISTERE), for three main reasons:

- Across a document, the number of time-event pairs to consider is large, as they are quadratic in the number of time and event expressions.
- There are almost no practically useful linguistic models that can be applied for capturing inter-sentence relations.
- Defining inter-sentence features is complex: their non-optimal definition in a task such as TERE – where there is a rather high imbalance between positive and negative examples – results in underperforming machine learning models.

In this paper, we design novel supervised models for ISTERE based on a structural representation of the pairs of sentences that contain the target rela-

International Joint Conference on Natural Language Processing, pages 1330–1338,
Nagoya, Japan, 14-18 October 2013.

tion arguments. We define methods to deal with time-event relations, where the text fragment indicating the time expression, e.g., *the appointment took effect Nov. 13* of the example above, is separated from the main event, e.g., *succession* and *stepping down*. In particular, our representation is constituted by a pair of shallow syntactic trees (one for each sentence containing the relation arguments), where their nodes are enriched with semantic labels, i.e., EVENT and TIME. We rely on automatic feature engineering with structural kernels (see e.g., (Moschitti, 2008; Moschitti, 2009)) to feed the learning algorithm with meaningful patterns implicitly described by such a representation. Kernels are applied to our shallow syntactic representations of text resulting in a model robust to noise and easily adaptable to new domains and tasks, such as ISTERE.

We tested our models on Machine Reading and TimeBank datasets over three different configurations: (i) relation arguments both within the same sentence, (ii) relation arguments in different sentences and (iii) relation arguments both, within and across, sentences. Our experiments demonstrate that such approach is very promising, as it improves over the state of the art for ISTERE by up to 20% in F1.

In the remainder of the paper, Sec. 2 surveys the related work, Sec. 3 presents the previous state-of-the-art models for intra-sentence TERE also using structural kernels, Sec. 4 describes our new models for intra/inter TERE, Sec. 5 lays out the experiments and, finally, Sec. 6 discusses the results deriving our conclusions.

2 Related Work

The extraction of relations between entities has been a long-standing topic of research, with work spanning more than a couple of decades, e.g., ACE (Doddington et al., 2004) and MUC (Grishman and Sundheim, 1996).

In particular, sentence-level Relation Extraction (RE) has been typically modeled with supervised approaches, using manually annotated data, such as ACE (Kambhatla, 2004). Most work has focused on kernel methods, i.e., string and tree kernels (Bunescu and Mooney, 2005; Culotta and Sorensen, 2004; Zhang et al., 2005; Zhang et al., 2006) or their combinations (Nguyen et al., 2009). From the kernel perspective, our approach to TERE is another variant of the general RE work using kernel: we use PTK applied to two-level shallow syntactic trees, which extracts a sort of hierarchical subsequences. This follows up our rather long research, e.g., tree kernels for modeling the relations between syntactic constituents embedded in pairs of text (i.e., question and answer passage) for answer re-ranking (Moschitti et al., 2007; Moschitti, 2008; Moschitti, 2009; Moschitti and Quarteroni, 2008; Moschitti and Quarteroni, 2010). A more computationally expensive solution based on enumerating relational links between constituents was given in (Zanzotto and Moschitti, 2006; Zanzotto et al., 2009) for the textual entailment task. Some faster versions were provided in (Moschitti and Zanzotto, 2007; Zanzotto et al., 2010). More efficient solutions based on a shallow tree and relational tags were recently proposed in (Severyn and Moschitti, 2012; Severyn et al., 2013).

Regarding the more specific task of extraction of temporal relations, the typical approaches follow similar principles of the above RE methods. Early work was devoted to ordering events with respect to one another, e.g., (Chambers and Jurafsky, 2008), and detecting their typical durations, e.g., (Pan et al., 2006). The TempEval workshops (Verhagen et al., 2007) defined the task of (i) extracting temporal relations between events and time expressions and (ii) naming relations like BEFORE, AFTER or OVERLAP. We focus on the first part of the TempEval task, following (Filatova and Hovy, 2001; Boguraev and Ando, 2005; Hovy et al., 2012), where we used the the system and results associated with the latter paper as a baseline of this paper. (Mirroshandel et al., 2011) used syntactic tree kernels for event-time links in the same sentence. As we aim at exploring long-distance RE, we consider more robust representations than syntactic trees, i.e., shallow syntactic trees, which we have successfully used in other research, e.g., (Severyn and Moschitti, 2012).

A recent challenge, i2b2[1] 2012, also dealing with ISTERE was carried out in the biomedical domain. We could not directly compare with the challenge's systems as their results were not available to us during the writing of this paper. Thus, we can only report on work targeting similar tasks, e.g., (Mani et al., 2006) used time relations between events to build a classifier that marks each pair of events with a temporal relation, exploiting temporal closure properties; and (ii) (Kolomiyets

[1] https://www.i2b2.org/pubs/index.html

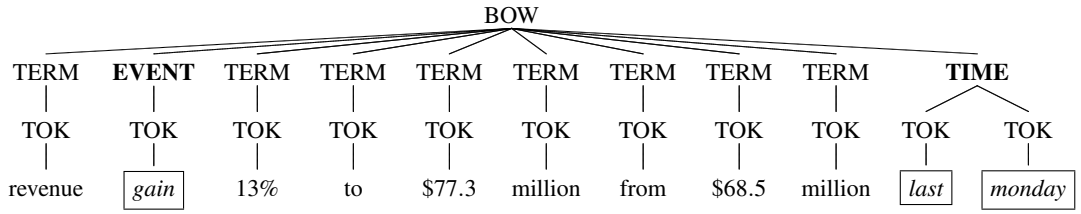

Figure 1: A kernel representations of the baseline model constituted by a bag-of-word (BOW) tree

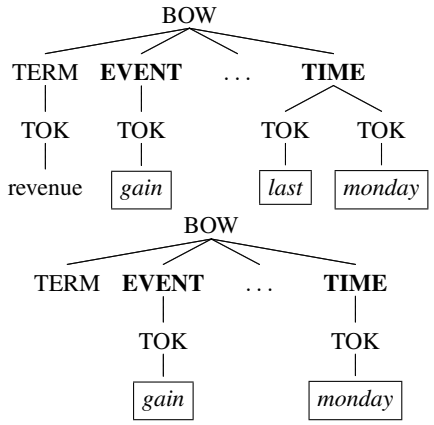

Figure 2: E.g. features from STK with BOW tree

et al., 2012) proposed to link events via partial ordering relations like BEFORE, AFTER, OVERLAP and IDENTITY.

Finally, a recent work explicitly tackling the ISTERE task is described in (Do et al., 2012). Their system was based on three classifiers: (i) a local classifier, which processes all pairs of events and time expressions in a document and decides which pairs are linked together; (ii) a classifier between pairs of events, which determines their relations: BEFORE, AFTER, OVERLAP and NO RELATION; and (iii) a joint model which, exploiting global constraints, can highly improve the overall ISTERE accuracy. We will further comment on this work in Sec. 6.

3 Baseline Models for TERE

The analysis of previous work has shown that there is almost no models for ISTERE. Therefore to align with prior work, we compare with previous models on *standard* TERE (extraction within a sentence). For this purpose, after formally defining the task, we describe the system we used as our baseline, which includes two types of features: (i) those manually designed also called linear features and (ii) structural features generated by tree kernels.

Task Definition. TERE is formally defined as follows: given the sets of expressions E denoting events or relation mentions, and T describing time expressions in the same document: (i) build all pairs $\langle e, t \rangle$ where $e \in E$ and $t \in T$; and (ii) classify $\langle e, t \rangle$ to determine if a time-event relation is held, i.e., if e occurs or holds within the temporal context specified by t. In our study, we assume that: (i) a timestamp must be explicitly stated for each event/relation that we consider to be in a temporal relation; and (ii) every event/relation is associated with only one time expression whereas a temporal expression can be linked to one or more events or relations.

Feature Vectors. We used system and features defined in our previous work (Hovy et al., 2012), which in turn are based on the work by (Boguraev and Ando, 2005). The feature set can be divided in three different classes: (i) Features associated with events or relations. These are very similar to features typically used to represent the context of entities in traditional relation extraction tasks, which are primarily syntactic features drawn from the parser and for reporting verbs. (ii) Features specific to the temporal expressions. These are primarily designed to capture various properties of the temporal expressions. For instance, whether it is a duration, time or date, or whether its premodifiers are among those that indicate the type of expression. We include also surface features, such as numeric or non-numeric tokens in the phrase. (iii) Features describing context around both the arguments. These are primarily drawn from the work by Boguraev and Ando, and include features such as n-grams and syntactic/structural patterns. The latter also cover syntactic relations between an event and a temporal expression, ordering of the two in the sentence, etc.

Tree Kernels. Convolution tree kernels (TK) compute the number of common substructures between two trees without explicitly considering the whole fragment space. TKs are equivalent to a scalar product between vectors of tree fragments. Therefore using TK in SVMs is equivalent to use subtrees as features. Different TKs exist, here we consider the partial tree kernel (PTK) defined in

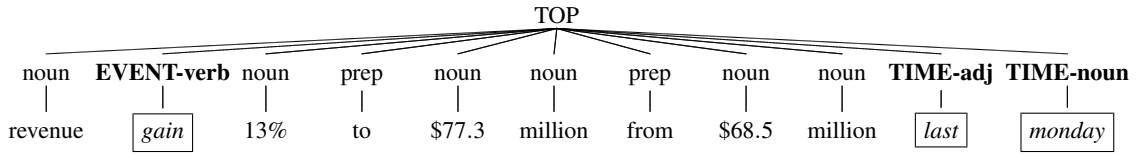

Figure 3: New sentence tree representation (STR)

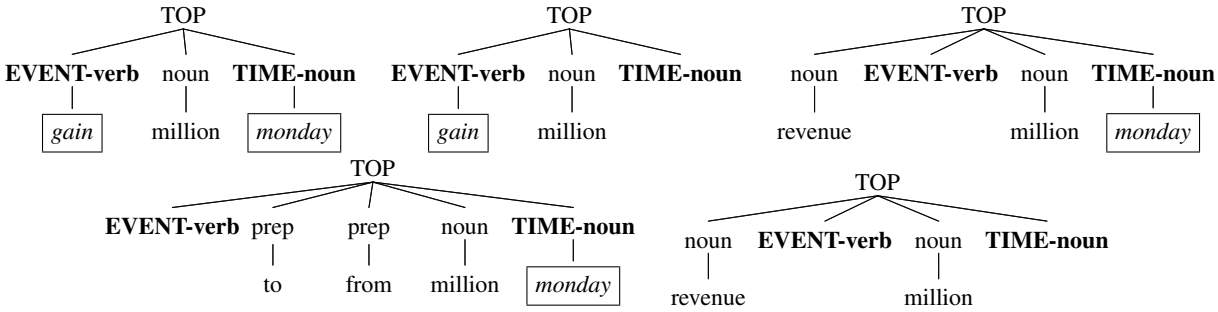

Figure 4: Some of the structural features generated by PTK when applied to the STR of Fig. 3

(Moschitti, 2006), which can count any tree fragments constituted by connected nodes.

Computational Structures. In (Hovy et al., 2012), we showed that combining manually engineered features with tree kernels produces a better model. We also show that exploiting syntactic information of the sentence containing the relational expressions is not trivial. Thus, we developed a bag-of-words (BOW) tree representation capturing the context of the target time/event expressions. The latter were marked-up with labels such as EVENT (or equivalently RELATION) and TIME. Figure 1 illustrates an example of the tree representations for the sentence:

Revenue gained 13% to $77.3 million from $68.5 million last Monday.

Such BOW tree is constituted by: (i) a root node; (ii) a conceptual level, which specifies the semantic type in the sentence, i.e., TERM, TIME or EVENT expressions; (iii) a token node level (TOK); and the lexical level, listing the words of the constituents. PTK applied to such trees generates features, such as `[TIME [last][monday]]` and `[TIME [[monday]]` or more interesting features like those shown in Figure 2. For example, such features can learn the pattern: `revenue, EVENT_gain, *, last, TIME`.

It should be noted that such a tree represents only one relation. In case a sentence contains more than one event/relation, separate trees for each must be generated (each will be a separate training/test instance). Such trees differ in the position of the EVENT/RELATION nodes (at level 1 of the tree).

Finally, in (Hovy et al., 2012) we showed that this model significantly improves over manually engineered features. However, to exploit syntactic information, we defined another separate tree with POS-tag nodes in place of words causing the features coming from different trees to be disjoint. This model cannot be applied for ISTERE as the large number of identical nodes, `TERM` and `TOK` would cause the PTK computational complexity to degenerate to $O(n^2)$ (see (Moschitti, 2009)).

4 Models for Inter-Sentence TERE

We describe here our new representation, which is an improvement over our previous models on intra-sentence TERE and, more importantly, can be used for ISTERE.

Intra-Sentence Representation. We improve on the previous work by reformulating the BOW tree as follows:

1. We remove the TERM and TOK levels and we propose only two levels — the POS-tag and the word sequences.

2. The annotation of the target time or event expression is directly performed on the POS-tag node.

For example, Fig. 3 shows the transformation of the trees in Fig. 1 to the new representation. The event *gain* and the time expression *last monday* are marked at POS-tag level[2]. This also compacts the segmentation of time expressions. As a result, the application of PTK to the new sentence tree

[2]The POS tagset is the one used in the IBM Watson system (Ferrucci et al., 2010).

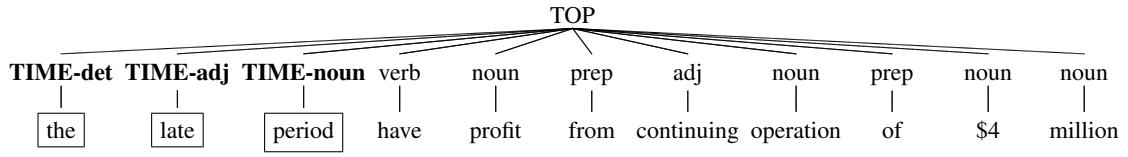

Figure 5: A second STR containing a time expression. Together with the one of Fig. 3 containing the event, it forms the representation pair, input to the kernel used for ISTERE.

representation (STR) generates very powerful and compact features, e.g., those described in Fig. 4. The last fragment of the figure suggests that if, in a sentence, there is the noun *revenue* followed by (i) any word tagged as EVENT, (ii) the noun *million* and (iii) any TIME expression, the EVENT is probably associated with such TIME expression. Note that this is not a rule but just a feature receiving a weight from SVMs based on training data.

Finally, it should be noted that the average running time of PTK will be $O(n)$, because the new tree does not contain many repeated node labels. This is far more efficient than the PTK applied to BOW trees (see (Moschitti, 2009)).

Inter-Sentence Representation. STR still cannot encode relations whose arguments are located in different sentences. Our approach for this problem is to design two STRs for the sentences containing the two potential arguments of a relation. For example, let us suppose that a time expression of the *gain*-EVENT in Fig. 3, e.g., *late period*, is expressed in the following sentence: *The late period has profit from continuing operations of $4 million.*

We produce another STR associated with it, as shown in Fig. 5. This way, we model $\langle e, t \rangle$ with a pair of trees $\langle E, T \rangle$, where $e \in E$ and $t \in T$ (see Sec. 3). Accordingly, we define the kernel $K(p_1, p_2)$ over two pairs $p_1 = \langle E_1, T_1 \rangle$ and $p_2 = \langle E_2, T_2, \rangle$ as: $TK(p_1, p_2) = \text{PTK}(E_1, E_2) + \text{PTK}(T_1, T_2)$.

It should be noted that: (i) the additive combination of kernels is still a valid kernel and it corresponds to the merged fragment space of E and T trees; (ii) the kernel product can also be applied but it has shown poor results in previous work (Moschitti, 2004); and (iii) PTK allows for modeling structural features in two sentences located in different part of the document. Thus, the features will be pairs of tree fragments from E and T. It is worth noting that the pairs of BOW and POS trees used in (Hovy et al., 2012) cause PTK to be too slow for this setting (although they may achieve comparable accuracy).

Additionally, the PTK can be combined with a linear kernel of manually engineered features using an additive operation. If $\vec{x}_i$ is the vector representation of the manually engineered features of p_i, the kernel combination of PTK and the linear kernel is $\text{PTK}(p_i, p_j) + \vec{x}_i \cdot \vec{x}_j$. The linear features extracted for the EVENT expression in one sentence and the TIME expression in the other sentence can reconstruct their shared context thanks to the pairs of tree fragments generated by PTK. The next section will empirically verify this hypothesis.

5 Experiments

In this section, (i) we compare our model against the state of the art for intra-sentence TERE; and (ii) we test its validity for ISTERE.

5.1 Setup

We used two corpora: Machine Reading Program (MRP) corpus to compare with our previous system (Hovy et al., 2012) (our baseline) and TimeBank data (Pustejovsky et al., 2003), which in contrast to MRP data, enables us to train and test our system with inter-sentence gold standard (GS) annotation. During testing, we used GS annotations for the timestamps and events, i.e., we only classify which events and time expressions are linked together.

MRP data. Following our work in (Hovy et al., 2012), we used the data made available in MRP related to linking timestamps and events in the intelligence community (IC) domain (Strassel et al., 2010). It is based on news reports about terrorism taken from the Gigaword corpus. It includes 169 documents containing $2,046$ pairs of event and temporal expressions (505 positive, $1,541$ negative instances) within the same sentence. We increased the original number of event instances by means of gold event-coreference annotations, i.e., two events that co-refer will share their annotated time expressions; thus we can merge them and increase the size of our gold annotation. As before, 41% of all correct event-time pairs are not in the same sentence (for relations this ratio is more than 80% of the correct fluent-time links).

	Training Set			Validation Set			Test Set		
	Pos.	Neg.	Total	Pos.	Neg.	Total	Pos.	Neg.	Total
DIST=0	1,125	2,754	3,879	155	405	560	162	463	625
DIST>0	900	65,221	66,121	129	9,311	9,440	182	16,600	16,782
DIST≥0	2,025	67,975	70,000	284	9,716	10,000	344	17,063	17,407

Table 1: Distribution of data in the three TimeBank subsets.

While it allows us to compare with previous work, the MRP data is not very well suited for training and testing new systems modeling ISTERE. Indeed, almost all inter-sentence pairs of time/event contain members that (i) are coreference of either the time or the event and (ii) are typically located in the same sentence. This means that, given an accurate system for intra-sentence TERE and a good coreference resolution system, ISTERE described in the MRP corpus can be easily solved. The combined system is still very complex and interesting, but it inevitably falls in the class of coreference resolution problems. Here we aim at studying linguistic phenomena directly connected to inter-sentence relations, which go beyond coreference resolution. For this reason, we also ran experiments on a second corpus described below, which is more suitable to our study.

TimeBank corpus. Distributed by the Linguistic Data Consortium[3], it consists of 183 documents – news articles from several news sources that have been annotated with event and time expressions compliant with the TimeML specification[4]. We divided the corpus in three subsets containing relations whose arguments are located in: (i) the same sentence (DIST=0), (ii) more than one sentence (DIST>0) and (iii) both previous cases (DIST≥0). The distribution of positive and negative examples in the training, validation and test sets are reported in Table 1. It is interesting to note that the distribution of positive examples in DIST=0, i.e., the intra-sentence relations, is 30% of all possible pairs. In contrast, such distribution in DIST>0, i.e., inter-sentence relations, drastically reduces to 1.4% of the pairs occurring in a document. This imbalance immediately gives the feeling of the complexity of the ISTERE task.

Learning Model. we used SVM-Light-TK (Moschitti, 2006; Joachims, 1999), which enables the use of the Partial Tree Kernel (PTK) (Moschitti, 2006). We used the default kernel hyperparameters and the margin/error trade-off parameter to favor replicability of our results, and study instead the cost-factor parameter as it tunes the balance between Precision and Recall, which is very critical for our task (highly skewed datasets).

Measures. We estimated Precision, Recall and F1 with 10-fold cross-validation for MRP experiments for comparing with (Hovy et al., 2012). For TimeBank, we drew F1 plots using the random test set described in Table 1. To estimate the final F1 of our models, we used 3-fold cross-validation[5] applied to the merged train and test set in the table. In this case the cost-factors were estimated from the validation set (see the table above), which is not part of the merged data used for the 3-fold cross-validation. It should be noted that to create the folds and the other subsets, we took care to not mix RE pairs between the folds or between training, validation and test set.

5.2 Intra-Sentence TERE: MRP Results

We trained SVMs using PTK applied to STR of single sentences and also combined with linear features. We tested a few parameter values of the Precision/Recall trade-off (cost-factor) on a validation set then, following (Hovy et al., 2012), we ran 10-fold cross-validation. The average F1 was 76.84, which is directly comparable with the outcome we reported in (Hovy et al., 2012), i.e., an F1 of 76.5 (when linear features are used in combination with tree kernels). It should be noted that: (i) in (Hovy et al., 2012) we showed that our system achieved the same accuracy than the best system of TempEval-2; and (ii) our STR provides the same results of the combination of the two structures we used in the model above.

Additionally, we tested PTK alone and attained an F1 of 74.45. This basically suggests that if we only use tree kernels, we can trade-off several months of work for manual feature engineering for a little bit less accurate system. Those unfamiliar with structural kernels may think that the time spent for engineering tree representations is comparable to the one spent for engineering features.

[3] LDC2006T08 at `http://www.timeml.org/site/timebank/documentation-1.2.html`

[4] The inter-annotator agreement numbers are specified in the referring website

[5] It is more suitable than a 10-fold setting for deriving the final accuracy, given the very low number of positive examples in DIST>0.

However, the abstraction provided by the tree kernels suggests that the effort required in engineering trees is orders of magnitudes lower. The baseline system using the manually engineered features was designed at IBM and required several months of manual effort to engineer, code and tune features. Our expert on kernel methods (who is not an expert on TERE) modeled STR in 20 minutes and the implementation only concerned the construction of strings representing trees like those in fig. 3 and 5. While this is anecdotal evidence, it is a good indicator of the power of tree kernels.

Furthermore, our experiments show that the combination of tree kernels and feature vectors is much more adaptable to variations of the TERE task and data. This can be observed, for instance, when considering RE from TimeBank.

5.3 Cost-factor role in ISTERE: TimeBank

We performed the first experiment using DIST=0 data and three different models, Linear (i.e., only using the manual features), PTK (which in this case only uses one tree) and their additive combination, i.e., Linear + PTK. In these experiments, the only critical parameter is the one tuning the Precision/Recall trade off (cost-factor parameter) as the high data imbalance between negative and positive examples can result in imbalanced Precision and Recall. Thus, we plotted the F1 of the above models (derived on the test set) according to a reasonable set of values of such parameter. The result is shown in Fig. 6. We note that (i) PTK produces a better F1 than linear features for any parameter value; (ii) the combination Linear + PTK highly improves on both achieving an interesting F1 of 64.35; (iii) in comparison with MRP, where the best model achieves an F1 of 76.84, the TimeBank task appears to be more difficult.

We ran an experiment for DIST>0, which considers only inter-sentence relations. Fig. 7 shows a similar curve as before, except that manual features have a higher accuracy than PTK. The F1, however, is rather low, indicating the complexity of the task and the inadequacy of manual features.

As predicted in Sec. 4, the combination of inter-sentence structural and manual features highly improves on the system F1 achieving a state-of-the-art value of 38.82 for ISTERE. Although, the result does not still guarantee a successful use of the proposed model for real-world applications, it clearly shows a promising research direction.

Finally, we tested DIST≥0, i.e., the complete

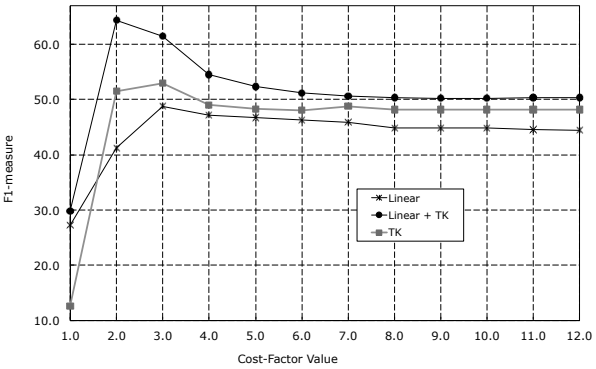

Figure 6: TERE cost-factor impact on DIST=0

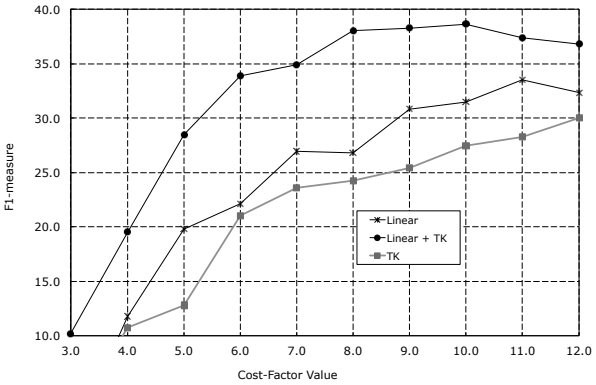

Figure 7: TERE cost-factor impact on DIST>0

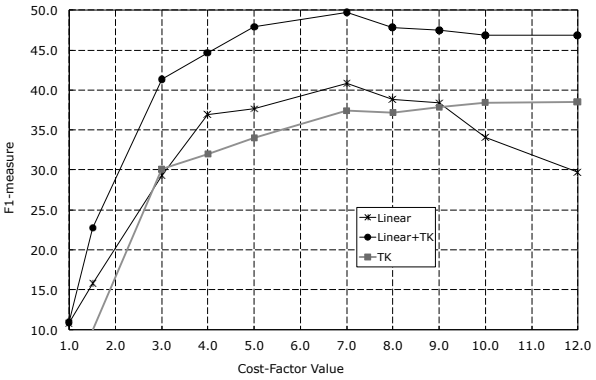

Figure 8: TERE cost-factor impact on DIST≥ 0

TERE task on entire documents. Fig. 8 shows comparable performance between PTK and Linear but again when combined together the improvement is rather high, i.e., up to 10 absolute percent points (25% of relative improvement on manual features). It should be noted that the above results do not indicate the final system accuracy: for this purpose, the next section shows the 3-fold cross-validation results using cost-fact values that are (i) derived from the validation set (not included in the cross-validation data) and (ii) slightly different from those optimizing the plot in the figures.

	DIST=0			DIST>0			DIST≥0		
	Linear	PTK	Lin.+PTK	Linear	PTK	Lin.+PTK	Linear	PTK	Lin.+PTK
Prec.	36.7±1.3	57.2±0.7	60.9±3.0	18.5±2.5	29.1±5.5	29.2±4.4	33.8±1.1	34.2±2.5	39.8±2.3
Rec.	89.4±3.2	42.4±1.9	64.0±2.9	55.5±5.0	20.6±8.2	32.3±7.9	57.7±1.1	46.7±1.5	58.4±4.0
F1	52.0±0.9	48.7±1.0	62.3±1.7	27.7±3.4	23.1±5.4	30.2±4.5	42.6±1.0	39.4±1.2	47.2±1.8

Table 2: 3-fold cross-validation results for DIST=0, DIST>0 and DIST≥0 tasks.

5.4 Cross-Validation Results

The previous section demonstrates the superiority of the combined Linear + PTK model over manual features for any value of the cost-factor parameter. To assess the significance of this, we carried out 3-cross-fold cross validation. Table 2 shows the average Precision, Recall and F1 over the 3-folds along with the associated standard deviation (preceded by ±). We note that (i) the relative improvement over the Linear model derived on DIST=0, about 20%, confirms the results showed by the plots; and (ii) the relative improvement derived on DIST>0 and DIST≥0 is lower, although still remarkable, i.e., up to 9% and 12%, respectively. This is probably due to the fact that 2 folds constitute a training set of 58K instances (for DIST≥0), whereas in the plot experiments the training data contained 70K examples. Evidently the more complex patterns needed for long-distance TERE require more training data to express their entire potential. Finally, feature vectors perform better than structural kernels alone but this does not contradict the fact that kernels save potentially large engineering work since: (i) even if the features had been engineered for MRP and used for TimeBank, the effort would have been done in any case whereas kernels almost completely avoid it; and (ii) the combination largely improve on the feature vectors: this may avoid the need of additional work for feature refining.

6 Discussion and Conclusions

Previous work has proposed intra-sentence TERE models based on manually designed features and tree kernels. In this paper, we propose new models for inter-and intra-sentence TERE. We provided a flexible kernel, which improves efficiency and capacity of generating meaningful features. It can be applied to the pairs of all document sentences for modeling ISTERE. This enables the use of all possible pairs of tree fragments from the time and event sentences as features, which improve on the features manually designed in (Hovy et al., 2012). For example, the latter cannot capture the relation with the "document date", when the time expressions occur in the titles. This kind of features can be added but a study of the problem and engineering effort are required. In contrast, our models can automatically generate such features.

Our experiments on MRP and TimeBank show that our approach provides high accuracy, up to 20% of relative improvement over state of the art. The reason for such impressive results is the adaptability and automatic feature engineering properties of tree kernels. Indeed, new data and settings pose new challenges to the RE systems, which require effort in engineering both features and methods. Our approach alleviates such effort as we can use a more general-purpose technology.

In this work, our model has been applied to establish the link between time expressions and events. However, in general, our model could be applied to the complete TERE task, thus also determining the relation types. Interestingly, the model proposed in (Do et al., 2012) is based on the pairwise classifiers we study in this paper. Although, the authors used a different dataset[6], which makes an exact comparison with their systems difficult, we note that their local pair classifiers achieved an F1 of 42.13 (no global model, so the same setting as ours) and an F1 of 46.01 using their global model based on ILP. Our local pairwise classifier attained an F1 of 47.2, which can be used as input to the global model to further boost the overall system accuracy.

Finally, the pairwise approach may be considered computationally expensive. However, with modern technology, $O(n^2)$ complexity (where n is the number of sentences in a document) is feasible. PTK is efficient and can be made faster with recent reverse kernel engineering work (Pighin and Moschitti, 2010; Pighin and Moschitti, 2009).

In summary, the main message of this paper is that ISTERE is complex, requiring a significant engineering effort. We have shown that tree kernels are adaptable, requiring less effort and improving on the state of the art in the full TERE task – relations spanning more than one sentence.

[6]They annotated a portion of the ACE corpus with (i) event mention and time interval association, and (ii) the temporal relations between event mentions.

Acknowledgements

This research is partially supported by the EU's 7th Framework Program (FP7/2007-2013) (#288024 LIMOSINE project) and an Open Collaborative Research (OCR) award from IBM Research.

References

Branimir Boguraev and Rie Kubota Ando. 2005. Timeml-compliant text analysis for temporal reasoning. In *Proceedings of IJCAI*.

R. Bunescu and R. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of HLT-EMNLP*.

N. Chambers and D. Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*.

A. Culotta and J. Sorensen. 2004. Dependency tree kernels for relation extraction. In *ACL*.

Q. Do, W. Lu, and D. Roth. 2012. Joint inference for event timeline construction. In *the joint EMNLP and CoNLL conference*.

G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. 2004. The automatic content extraction program – tasks, data and evaluation. In *LREC*.

D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, N. Schlaefer, and C. Welty. 2010. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3).

E. Filatova and E. Hovy. 2001. Assigning time-stamps to event-clauses. In *the workshop on Temporal and spatial information processing*.

R. Grishman and B. Sundheim. 1996. Message Understanding Conference - 6: A Brief History. In *Coling*.

D. Hovy, J. Fan, A. Gliozzo, S. Patwardhan, and C. Welty. 2012. When did that happen? – linking events and relations to timestamps. In *EACL*.

T. Joachims. 1999. Making large-scale SVM learning practical. *Advances in Kernel Methods – Support Vector Learning*, 13.

N. Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *ACL*.

O. Kolomiyets, S. Bethard, and M.-F. Moens. 2012. Extracting narrative timelines as temporal dependency structures. In *ACL*.

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

S. A. Mirroshandel, M. Khayyamian, and G. Ghassem-Sani. 2011. Syntactic tree kernels for event-time temporal relation learning. *HLT*.

A. Moschitti and S. Quarteroni. 2008. Kernels on Linguistic Structures for Answer Extraction. In *ACL*.

A. Moschitti and S. Quarteroni. 2010. Linguistic Kernels for Answer Re-ranking in Question Answering Systems. *Information Processing & Management*.

A. Moschitti and F. M. Zanzotto. 2007. Fast and effective kernels for relational learning from texts. In *ICML*.

A. Moschitti, S. Quarteroni, R. Basili, and S. Manandhar. 2007. Exploiting syntactic and shallow semantic kernels for question/answer classification. In *ACL*.

A. Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *ACL*.

A. Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML*.

A. Moschitti. 2008. Kernel methods, syntax and semantics for relational text categorization. In *CIKM*.

A. Moschitti. 2009. Syntactic and Semantic Kernels for Short Text Pair Categorization. In *EACL*.

T.-V. T. Nguyen, A. Moschitti, and G. Riccardi. 2009. Convolution kernels on constituent, dependency and sequential structures for relation extraction. In *EMNLP*.

F. Pan, R. Mulkar, and J. R. Hobbs. 2006. Learning event durations from event descriptions. In *Coling-ACL*.

D. Pighin and A. Moschitti. 2009. Reverse engineering of tree kernel feature spaces. In *EMNLP*.

D. Pighin and A. Moschitti. 2010. On reverse feature engineering of syntactic tree kernels. In *CoNLL*.

J. Pustejovsky, P. Hanks, R. Saurí, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. The TIMEBANK Corpus.

A. Severyn and A. Moschitti. 2012. Structural relationships for large-scale learning of answer reranking. In *SIGIR*.

A. Severyn, M. Nicosia, and A. Moschitti. 2013. Learning adaptable patterns for passage reranking. In *CoNLL*.

S. Strassel, D. Adams, H. Goldberg, J. Herr, R. Keesing, D. Oblinger, H. Simpson, R. Schrag, and J. Wright. 2010. The DARPA MRP. In *LREC*.

M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *the Workshop on Sem. Ev.*.

M. Verhagen, R. Sauri, T. Caselli, and J. Pustejovsky. 2010. Semeval-2010 task 13: Tempeval-2. In *the Workshop on Sem. Evaluation*.

F. M. Zanzotto and A. Moschitti. 2006. Automatic Learning of Textual Entailments with Cross-Pair Similarities. In *COLING-ACL*.

F. M. Zanzotto, M. Pennacchiotti, and A. Moschitti. 2009. A Machine Learning Approach to Recognizing Textual Entailment. *JNLE*.

F. M. Zanzotto, L. Dell'Arciprete, and A. Moschitti. 2010. Efficient graph kernels for textual entailment recognition. *Fundamenta Informaticae*, 2010.

M. Zhang, J. Su, D. Wang, G. Zhou, and C. L. Tan. 2005. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Proc. of IJCNLP*.

M. Zhang, J. Zhang, J. Su, and G. Zhou. 2006. A composite kernel to extract relations between entities with both flat and structured features. In *COLING-ACL*.