When Did *that* Happen? — Linking Events and Relations to Timestamps

Dirk Hovy*, **James Fan**, **Alfio Gliozzo**, **Siddharth Patwardhan** and **Chris Welty**
IBM T. J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
dirkh@isi.edu, {fanj,gliozzo,siddharth,welty}@us.ibm.com

Abstract

We present work on linking events and fluents (i.e., relations that hold for certain periods of time) to temporal information in text, which is an important enabler for many applications such as timelines and reasoning. Previous research has mainly focused on temporal links for events, and we extend that work to include fluents as well, presenting a common methodology for linking both events and relations to timestamps within the same sentence. Our approach combines tree kernels with classical feature-based learning to exploit context and achieves competitive F1-scores on event-time linking, and comparable F1-scores for fluents. Our best systems achieve F1-scores of 0.76 on events and 0.72 on fluents.

1 Introduction

It is a long-standing goal of NLP to process natural language content in such a way that machines can effectively reason over the entities, relations, and events discussed within that content. The applications of such technology are numerous, including intelligence gathering, business analytics, healthcare, education, etc. Indeed, the promise of *machine reading* is actively driving research in this area (Etzioni et al., 2007; Barker et al., 2007; Clark and Harrison, 2010; Strassel et al., 2010).

Temporal information is a crucial aspect of this task. For a machine to successfully understand natural language text, it must be able to associate time points and temporal durations with relations and events it discovers in text.

In this paper we present methods to establish links between events (e.g. "bombing" or "election") or fluents (e.g. "spouseOf" or "employedBy") and temporal expressions (e.g. "last Tuesday" and "November 2008"). While previous research has mainly focused on temporal links for events only, we deal with both events and fluents with the same method. For example, consider the sentence below

> *Before his death in October, Steve Jobs led Apple for 15 years.*

For a machine reading system processing this sentence, we would expect it to link the fluent *CEO_of*(Steve_Jobs, Apple) to time duration "15 years". Similarly we expect it to link the event "death" to the time expression "October".

We do not take a strong "ontological" position on what events and fluents are, as part of our task these distinctions are made a priori. In other words, events and fluents are input to our temporal linking framework. In the remainder of this paper, we also do not make a strong distinction between relations in general and fluents in particular, and use them interchangeably, since our focus is only on the specific types of relations that represent fluents. While we only use binary relations in this work, there is nothing in the framework that would prevent the use of n-ary relations. Our work focuses on accurately identifying temporal links for eventual use in a machine reading context.

In this paper, we describe a single approach that applies to both fluents and events, using feature engineering as well as tree kernels. We show that we can achieve good results for both events and fluents using the same feature space, and advocate

*The first author conducted this research during an internship at IBM Research.

Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 185–193,
Avignon, France, April 23 - 27 2012.

the versatility of our approach by achieving competitive results on yet another similar task with a different data set.

Our approach requires us to capture contextual properties of text surrounding events, fluents and time expressions that enable an automatic system to detect temporal linking within our framework. A common strategy for this is to follow standard feature engineering methodology and manually develop features for a machine learning model from the lexical, syntactic and semantic analysis of the text. A key contribution of our work in this paper is to demonstrate a shallow tree-like representation of the text that enables us to employ *tree kernel* models, and more accurately detect temporal linking. The feature space represented by such tree kernels is far larger than a manually engineered feature space, and is capable of capturing the contextual information required for temporal linking.

The remainder of this paper goes into the details of our approach for temporal linking, and presents empirical evidence for the effectiveness of our approach. The contributions of this paper can be summarized as follows:

1. We define a common methodology to link events and fluents to timestamps.

2. We use tree kernels in combination with classical feature-based approaches to obtain significant gains by exploiting context.

3. Empirical evidence illustrates that our framework for temporal linking is very effective for the task, achieving an F1-score of 0.76 on events and 0.72 on fluents/relations, as well as 0.65 for TempEval2, approaching state-of-the-art.

2 Related Work

Most of the previous work on relation extraction focuses on entity-entity relations, such as in the ACE (Doddington et al., 2004) tasks. Temporal relations are part of this, but to a lesser extent. The primary research effort in event temporality has gone into ordering events with respect to one another (e.g., Chambers and Jurafsky (2008)), and detecting their typical durations (e.g., Pan et al. (2006)).

Recently, TempEval workshops have focused on the temporal related issues in NLP. Some of the TempEval tasks overlap with ours in many ways. Our task is similar to task A and C of TempEval-1 (Verhagen et al., 2007) in the sense that we attempt to identify temporal relation between events and time expressions or document dates. However, we do not use a restricted set of events, but focus primarily on a single temporal relation t_{link} instead of named relations like BEFORE, AFTER or OVERLAP (although we show that we can incorporate these as well). Part of our task is similar to task C of TempEval-2 (Verhagen et al., 2010), determining the temporal relation between an event and a time expression in the same sentence. In this paper, we do apply our system to TempEval-2 data and compare our performance to the participating systems.

Our work is similar to that of Boguraev and Ando (2005), whose research only deals with temporal links between events and time expressions (and does not consider relations at all). They employ a sequence tagging model with manual feature engineering for the task and achieved state-of-the-art results on Timebank (Pustejovsky et al., 2003) data. Our task is slightly different because we include relations in the temporal linking, and our use of tree kernels enables us to explore a wider feature space very quickly.

Filatova and Hovy (2001) also explore temporal linking with events, but do not assume that events and time stamps have been provided by an external process. They used a heuristics-based approach to assign temporal expressions to events (also relying on the proximity as a base case). They report accuracy of the assignment for the correctly classified events, the best being 82.29%. Our best event system achieves an accuracy of 84.83%. These numbers are difficult to compare, however, since accuracy does not efficiently capture the performance of a system on a task with so many negative examples.

Mirroshandel et al. (2011) describe the use of syntactic tree kernels for event-time links. Their results on TempEval are comparable to ours. In contrast to them, we found, though, that syntactic tree kernels alone do not perform as well as using several flat tree representations.

3 Problem Definition

The task of linking events and relations to time stamps can be defined as the following: given a set of expressions denoting events or relation men-

tions in a document, and a set of time expressions in the same document, find all instances of the t_{link} relation between elements of the two input sets. The existence of a $t_{link}(e, t)$ means that e, which is an event or a relation mention, occurs within the temporal context specified by the time expression t.

Thus, our task can be cast as a binary relation classification task: for each possible pair of (event/relation, time) in a document, decide whether there exists a link between the two, and if so, express it in the data.

In addition, we make these assumptions about the data:

1. There does not exist a timestamp for every event/relation in a document. Although events and relations typically have temporal context, it may not be explicitly stated in a document.

2. Every event/relation has at most one time expression associated with it. This is a simplifying assumption, which in the case of relations we explore as future work.

3. Each temporal expression can be linked to one or more events or relations. Since multiple events or relations may happen for a given time, it is safe to assume that each temporal expression can be linked to more than one event/relation.

In general, the events/relations and their associated timestamps may occur within the same sentence or may occur across different sentences. In this paper, we focus on our effort and our evaluation on the same sentence linking task.

In order to solve the problem of temporal linking completely, however, it will be important to also address the links that hold between entities across sentences. We estimate, based on our data set, that across sentence links account for 41% of all correct event-time pairs in a document. For fluents, the ratio is much higher, more than 80% of the correct fluent-time links are across sentences. One of the main obstacles for our approach in the cross-sentence case is the very low ratio of positive to negative instances ($3 : 100$) in the set of all pairs in a document. Most pairs are not linked to one another.

4 Temporal Linking Framework

As previously mentioned, we approach the temporal linking problem as a classification task. In the framework of classification, we refer to each pair of (*event/relation, temporal expression*) occurring within a sentence as an instance. The goal is to devise a classifier that separates positive (i.e., linked) instances from negative ones, i.e., pairs where there is no link between the event/relation and the temporal expression in question. The latter case is far more frequent, so we have an inherent bias toward negative examples in our data.[1]

Note that the basis of the positive and negative links is the context around the target terms. It is impossible even for humans to determine the existence of a link based only on the two terms without their context. For instance, given just two words (e.g., "said" and "yesterday") there is no way to tell if it is a positive or a negative example. We need the context to decide.

Therefore, we base our classification models on contextual features drawn from lexical and syntactic analyses of the text surrounding the target terms. For this, we first define a feature-based approach, then we improve it by using tree kernels. These two subsections, plus the treatment of fluent relations, are the main contributions of this paper. In all of this work, we employ SVM classifiers (Vapnik, 1995) for machine learning.

4.1 Feature Engineering

A manual analysis of development data provided several intuitions about the kinds of features that would be useful in this task. Based on this analysis and with inspiration from previous work (cf. Boguraev and Ando (2005)) we established three categories of features whose description follows.

Features describing events or relations. We check whether the event or relation is phrasal, a verb, or noun, whether it is present tense, past tense, or progressive, the type assigned to the event/relation by the UIMA type system used for processing, and whether it includes certain trigger words, such as reporting verbs ("said", "reported", etc.).

[1] Initially, we employed an instance filtering method to address this, which proved to be ineffective and was subsequently left out.

Features describing temporal expressions. We check for the presence of certain trigger words (*last, next, old,* numbers, etc.) and the type of the expression (DURATION, TIME, or DATE) as specified by the UIMA type system.

Features describing context. We also include syntactic/structural features, such as testing whether the relation/event dominates the temporal expression, which one comes first in the sentence order, and whether either of them is dominated by a separate verb, preposition, "that" (which often indicates a subordinate sentence) or counterfactual nouns or verbs (which would negate the temporal link).

It is not surprising that some of the most informative features (event comes before temporal expression, time is syntactic child of event) are strongly correlated with the baselines. Less salient features include the test for certain words indicating the event is a noun, a verb, and if so which tense it has and whether it is a reporting verb.

4.2 Tree Kernel Engineering

We expect that there exist certain patterns between the entities of a temporal link, which manifest on several levels: some on the lexical level, others expressed by certain sequences of POS tags, NE labels, or other representations. Kernels provide a principled way of expanding the number of dimensions in which we search for a decision boundary, and allow us to easily model local sequences and patterns in a natural way (Giuliano et al., 2009). While it is possible to define a space in which we find a decision boundary that separates positive and negative instances with manually engineered features, these features can hardly capture the notion of context as well as those explored by a tree kernel.

Tree Kernels are a family of kernel functions developed to compute the similarity between tree structures by counting the number of subtrees they have in common. This generates a high-dimensional feature space that can be handled efficiently using dynamic programming techniques (Shawe-Taylor and Christianini, 2004). For our purposes we used an implementation of the Subtree and Subset Tree (SST) (Moschitti, 2006).

The advantages of using tree kernels are two-fold: thanks to an existing implementation (SVM^{light} with tree kernels, Moschitti (2004)), it is faster and easier than traditional feature engineering. The tree structure also allows us to use different levels of representations (POS, lemma, etc.) and combine their contributions, while at the same time taking into account the ordering of labels. We use POS, lemma, semantic type, and a representation that replaces each word with a concatenation of its features (capitalization, countable, abstract/concrete noun, etc.).

We developed a shallow tree representation that captures the context of the target terms, without encoding too much structure (which may prevent generalization). In essence, our tree structure induces behavior somewhat similar to a string kernel. In addition, we can model the tasks by providing specific markup on the generated tree. For example, in our experiment we used the labels EVENT (or equivalently RELATION) and TIMESTAMP to mark our target terms. In order to reduce the complexity of this comparison, we focus on the substring between event/relation and time stamp and the rest of the tree structure is truncated.

Figure 1 illustrates an example of the structure described so far for both lemmas and POS tags (note that the lowest level of the tree contains tokenized items, so their number can differ form the actual words, as in "attorney_general"). Similar trees are produced for each level of representations used, and for each instance (i.e., pair of time expressions and event/relation). If a sentence contains more than one event/relation, we create separate trees for each of them, which differ in the position of the EVENT/RELATION marks (at level 1 of the tree).

The tree kernel implicitly expands this structure into a number of substructures allowing us to capture sequential patterns in the data. As we will see, this step provides significant boosts to the task performance.

Curiously, using a full-parse syntactic tree as input representation did not help performance. This is in line with our finding that syntactic relations are less important than sequential patterns (see also Section 5.2). Therefore we adopted the "string kernel like" representation illustrated in Figure 1.

Scores of supporters of detained Egyptian opposition leader Nur demonstrated outside the attorney general's office in Cairo last Saturday, demanding he be freed immediately.

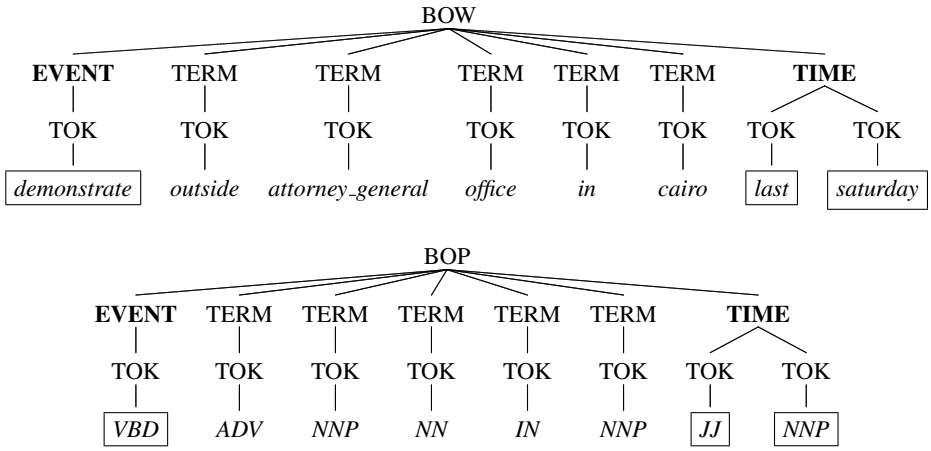

Figure 1: Input Sentence and Tree Kernel Representations for Bag of Words (BOW) and POS tags (BOP)

5 Evaluation

We now apply our models to real world data, and empirically demonstrate their effectiveness at the task of temporal linking. In this section, we describe the data sets that were used for evaluation, the baselines for comparison, parameter settings, and the results of the experiments.

5.1 Benchmark

We evaluated our approach in 3 different tasks:

1. Linking Timestamps and Events in the IC domain
2. Linking Timestamps and Relations in the IC domain
3. Linking Events to Temporal Expressions (TempEval-2, task C)

The first two data sets contained annotations in the intelligence community (IC) domain, i.e., mainly news reports about terrorism. It comprised 169 documents. This dataset has been developed in the context of the machine reading program (MRP) (Strassel et al., 2010). In both cases our goal is to develop a binary classifier to judge whether the event (or relation) overlaps with the time interval denoted by the timestamp. Success of this classification can be measured by precision and recall on annotated data.

We originally considered using accuracy as a measure of performance, but this does not correctly reflect the true performance of the system: given the skewed nature of the data (much smaller number of positive examples), we could achieve a high accuracy simply by classifying all instances as negative, i.e., not assigning a time stamp at all. We thus decided to report precision, recall and F1. Unless stated otherwise, results were achieved via 10-fold cross-validation (10-CV).

The number of instances (i.e., pairs of event and temporal expression) for each of the different cases listed above was (in brackets the ratio of positive to negative instances).

- events: 2046 (505 positive, 1541 negative)
- relations: 6526 (1847 positive, 4679 negative)

The size of the relation data set after filtering is 5511 (1847 positive, 3395 negative).

In order to increase the originally lower number of event instances, we made use of the annotated event-coreference as a sort of closure to add more instances: if events A and B corefer, and there is a link between A and time expression t, then there is also a link between B and t. This was not explicitly expressed in the data.

For the task at hand, we used gold standard annotations for timestamps, events and relations. The task was thus not the identification of these objects (a necessary precursor and a difficult task in itself), but the decision as to which events and time expressions could and should be linked.

We also evaluated our system on TempEval-2 (Verhagen et al., 2010) for better comparison

to the state-of-the-art. TempEval-2 data included the task of linking events to temporal expressions (there called "task C"), using several link types (OVERLAP, BEFORE, AFTER, BEFORE-OR-OVERLAP, OVERLAP-OR-AFTER). This is a bit different from our settings as it required the implementation of a multi-class classifier. Therefore we trained three different binary classifiers (using the same feature set) for the first three of those types (for which there was sufficient training data) and we used a one-versus-all strategy to distinguish positive from negative examples. The output of the system is the category with the highest SVM decision score. Since we only use three labels, we incur an error every time the gold label is something else. Note that this is stricter than the evaluation in the actual task, which left contestants with the option of skipping examples their systems could not classify.

5.2 Baselines

Intuitively, one would expect temporal expressions to be close to the event they denote, or even syntactically related. In order to test this, we applied two baselines. In the first, each temporal expression was linked to the closest event (as measured in token distance). In the second, we attached each temporal expression to its syntactic head, if the head was an event. Results are reported in Figure 2.

While these results are encouraging for our task, it seems at first counter-intuitive that the syntactic baseline does worse than the proximity-based one. It does, however, reveal two facts: events are not always synonymous with syntactic units, and they are not always bound to temporal expressions through direct syntactic links. The latter makes even more sense given that the links can even occur across sentence boundaries. Parsing quality could play a role, yet seems far fetched to account for the difference.

More important than syntactic relations seem to be sequential patterns on different levels, a fact we exploit with the different tree representations used (POS tags, NE types, etc.).

For relations, we only applied the closest-relation baseline. Since relations consist of two or more arguments that occur in different, often separated syntactic constituents, a syntactic approach seems futile, especially given our experience with events. Results are reported in Figure 3.

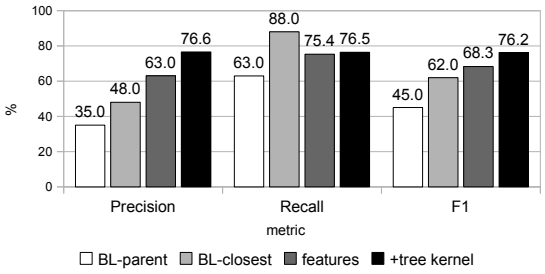

Figure 2: Performance on events

System	Accuracy
TRIOS	65%
this work	64.5%
JU-CSE, NCSU-indi TRIPS, USFD2	all 63%

Table 1: Comparison to Best Systems in TempEval-2

5.3 Events

Figure 2 shows the improvements of the feature-based approach over the two baseline, and the additional gain obtained by using the tree kernel. Both the features and tree kernels mainly improve precision, while the tree kernel adds a small boost in recall. It is remarkable, though, that the closest-event baseline has a very high recall value. This suggests that most of the links actually do occur between items that are close to one another. For a possible explanation for the low precision value, see the error analysis (Section 5.5).

Using a two-tailed t-test, we compute the significance in the difference between the F1-scores. Both the feature-based and the tree kernel approach improvements are statistically significant at $p < 0.001$ over the baseline scores.

Table 1 compares the performances of our system to the state-of-the-art systems on TempEval-2 Data, task C, showing that our approach is very competitive. The best systems there used sequential models. We attribute the competitive nature of our results to the use of tree kernels, which enables us to make use of contextual information.

5.4 Relations

In general, performance for relations is not as high as for events (see Figure 3). The reason here is two-fold: relations consist of two (or more) elements, which can be in various positions with respect to one another and the temporal expression, and each relation can be expressed in a number of

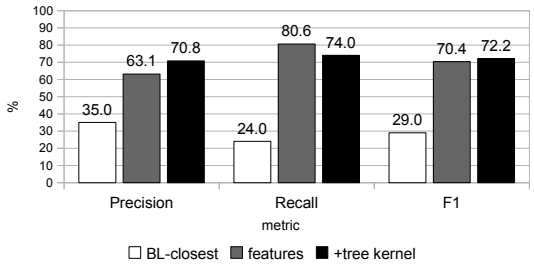

Figure 3: Performance on relations/fluents

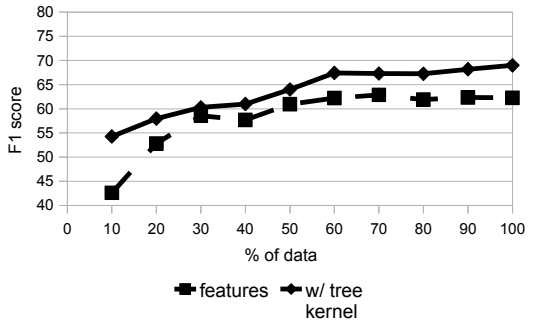

Figure 4: Learning curves for relation models

different ways.

Again, we perform significance tests on the difference in F1 scores and find that our improvements over the baseline are statistically significant at $p < 0.001$. The improvement of the tree kernel over the feature-based approach, however, are not statistically significant at the same value.

The learning curve over parts of the training data (exemplary shown here for relations, Figure 4)[2] indicates that there is another advantage to using tree kernels: the approach can benefit from more data. This is conceivably because it allows the kernel to find more common subtrees in the various representations the more examples it gets, while the feature space rather finds more instances that invalidate the expressiveness of features (i.e., it encounters positive and negative instances that have very similar feature vectors). The curve suggests that tree kernels could yield even better results with more data, while there is little to no expected gain using only features.

5.5 Error Analysis

Examining the misclassified examples in our data, we find that both feature-based and tree-kernel approaches struggle to correctly classify examples where time expression and event/relation are immediately adjacent, but unrelated, as in "the man arrested **last Tuesday told** the police ...", where *last Tuesday* modifies *arrested*. It limits the amount of context that is available to the tree kernels, since we truncate the tree representations to the words between those two elements. This case closely resembles the problem we see in the closest-event/relation baseline, which, as we have seen, does not perform too well. In this case, the incorrect event ("told") is as close to the time expression as the correct one ("arrested"), resulting in a false positive that affects precision. Features capturing the order of the elements do not seem help here, since the elements can be arranged in any order (i.e., temporal expression before or after the event/relation). The only way to solve this problem would be to include additional information about whether a time expression is already attached to another event/relation.

5.6 Ablations

To quantify the utility of each tree representation, we also performed all-but-one ablation tests, i.e., left out each of the tree representations in turn, ran 10-fold cross-validation on the data and observed the effect on F1. The larger the loss in F1, the more informative the left-out-representation. We performed ablations for both events and relations, and found that the ranking of the representations is the same for both.

In events and relations alike, leaving out POS trees has the greatest effect on F1, followed by the feature-bundle representation. Lemma and semantic type representation have less of an impact.

We hypothesize that the former two capture underlying regularities better by representing different words with the same label. Lemmas in turn are too numerous to form many recurring patterns, and semantic type, while having a smaller label alphabet, does not assign a label to every word, thus creating a very sparse representation that picks up more noise than signal.

In preliminary tests, we also used annotated dependency trees as input to the tree kernel, but found that performance improved when they were left out. This is at odds with work that clearly showed the value of syntactic tree kernels (Mirroshandel et al., 2011). We identify two potential causes—either our setup was not capable of correctly capturing and exploiting the information

[2]The learning curve for events looks similar and is omitted due to space constraints.

from the dependency trees, or our formulation of the task was not amenable to it. We did not investigate this further, but leave it to future work.

6 Conclusion and Future Work

We cast the problem of linking events and relations to temporal expressions as a classification task using a combination of features and tree kernels, with probabilistic type filtering. Our main contributions are:

- We showed that within-sentence temporal links for both events and relations can be approached with a common strategy.
- We developed flat tree representations and showed that these produce considerable gains, with significant improvements over different baselines.
- We applied our technique without great adjustments to an existing data set and achieved competitive results.
- Our best systems achieve F1 score of 0.76 on events and 0.72 on relations, and are effective at the task of temporal linking.

We developed the models as part of a machine reading system and are currently evaluating it in an end-to-end task.

Following tasks proposed in TempEval-2, we plan to use our approach for across-sentence classification, as well as a similar model for linking entities to the document creation date.

Acknowledgements

We would like to thank Alessandro Moschitti for his help with the tree kernel setup, and the reviewers who supplied us with very constructive feedback. Research supported in part by Air Force Contract FA8750-09-C-0172 under the DARPA Machine Reading Program.

References

Ken Barker, Bhalchandra Agashe, Shaw-Yi Chaw, James Fan, Noah Friedland, Michael Glass, Jerry Hobbs, Eduard Hovy, David Israel, Doo Soon Kim, Rutu Mulkar-Mehta, Sourabh Patwardhan, Bruce Porter, Dan Tecuci, and Peter Yeh. 2007. Learning by reading: A prototype system, performance baseline and lessons learned. In *Proceedings of the 22nd National Conference for Artificial Intelligence*, Vancouver, Canada, July.

Branimir Boguraev and Rie Kubota Ando. 2005. Timeml-compliant text analysis for temporal reasoning. In *Proceedings of IJCAI*, volume 5, pages 997–1003. IJCAI.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. pages 789–797. Association for Computational Linguistics.

Peter Clark and Phil Harrison. 2010. Machine reading as a process of partial question-answering. In *Proceedings of the NAACL HLT Workshop on Formalisms and Methodology for Learning by Reading*, Los Angeles, CA, June.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction program – tasks, data and evaluation. In *Proceedings of the LREC Conference*, Canary Islands, Spain, July.

Oren Etzioni, Michele Banko, and Michael Cafarella. 2007. Machine reading. In *Proceedings of the AAAI Spring Symposium Series*, Stanford, CA, March.

Elena Filatova and Eduard Hovy. 2001. Assigning time-stamps to event-clauses. In *Proceedings of the workshop on Temporal and spatial information processing*, volume 13, pages 1–8. Association for Computational Linguistics.

Claudio Giuliano, Alfio Massimiliano Gliozzo, and Carlo Strapparava. 2009. Kernel methods for minimally supervised wsd. *Computational Linguistics*, 35(4).

Seyed A. Mirroshandel, Mahdy Khayyamian, and Gholamreza Ghassem-Sani. 2011. Syntactic tree kernels for event-time temporal relation learning. *Human Language Technology. Challenges for Computer Science and Linguistics*, pages 213–223.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 335–es. Association for Computational Linguistics.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proceedings of EACL*, volume 6.

Feng Pan, Rutu Mulkar, and Jerry R. Hobbs. 2006. Learning event durations from event descriptions. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 393–400. Association for Computational Linguistics.

James Pustejovsky, Patrick Hanks, Roser Saurı, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa

Ferro, and Marcia Lazo. 2003. The TIMEBANK Corpus. In *Proceedings of Corpus Linguistics 2003*, pages 647–656.

John Shawe-Taylor and Nello Christianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Stephanie Strassel, Dan Adams, Henry Goldberg, Jonathan Herr, Ron Keesing, Daniel Oblinger, Heather Simpson, Robert Schrag, and Jonathan Wright. 2010. The DARPA Machine Reading Program-Encouraging Linguistic and Reasoning Research with a Series of Reading Tasks. In *Proceedings of LREC 2010*.

Vladimir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer, New York, NY.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75–80. Association for Computational Linguistics.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62. Association for Computational Linguistics.