Discovering Narrative Containers in Clinical Text

Timothy A. Miller[1], **Steven Bethard**[2], **Dmitriy Dligach**[1],
Sameer Pradhan[1], **Chen Lin**[1], and **Guergana K. Savova**[1]
[1] Children's Hospital Informatics Program, Boston Children's Hospital and Harvard Medical School
firstname.lastname@childrens.harvard.edu
[2] Center for Computational Language and Education Research, University of Colorado Boulder
steven.bethard@colorado.edu

Abstract

The clinical narrative contains a great deal of valuable information that is only understandable in a temporal context. Events, time expressions, and temporal relations convey information about the time course of a patient's clinical record that must be understood for many applications of interest. In this paper, we focus on extracting information about how time expressions and events are related by *narrative containers*. We use support vector machines with composite kernels, which allows for integrating standard feature kernels with tree kernels for representing structured features such as constituency trees. Our experiments show that using tree kernels in addition to standard feature kernels improves F1 classification for this task.

1 Introduction

Clinical narratives are a rich source of unstructured information that hold great potential for impacting clinical research and clinical care. These narratives consist of unstructured natural language descriptions of various stages of clinical care, which makes them information dense but challenging to use computationally. Information extracted from these narratives is already being used for clinical research tasks such as automatic phenotype classification for collecting disease cohorts retrospectively (Ananthakrishnan et al., 2013), which can in turn be used for a variety of studies, including pharmacogenomics (Lin et al., 2012; Wilke et al., 2011). Future applications may use information extracted from the clinical narrative at the point of care to assist physicians in decision-making in a real time fashion.

One of the most interesting and challenging aspects of clinical text is the pervasiveness of temporally grounded information. This includes a number of clinical concepts which are *events* with finite time spans (e.g., *surgery* or *x-ray*), *time expressions* (*December*, *postoperatively*), and links that relate events to times or other events. For example, *surgery last May* relates the time *last May* with the event *surgery* via the CONTAINS relation, while *Vicodin after surgery* relates the medication event *Vicodin* with the procedure event *surgery* via the AFTER relation. There are many potential applications of clinical information extraction that are only possible with an understanding of the ordering and duration of the events in a clinical encounter.

In this work we focus on extracting a particular temporal relation, CONTAINS, that holds between a time expression and an event expression. This level of representation is based on the computational discourse model of *narrative containers* (Pustejovsky and Stubbs, 2011), which are time expressions or events which are central to a section of a text, usually manifested by being relative hubs of temporal relation links. We argue that containment relations are useful as an intermediate level of granularity between full temporal relation extraction and "coarse" temporal bins (Raghavan et al., 2012) like *before admission*, *on admission*, and *after admission*. Correctly extracting CONTAINS relations will, for example, allow for more accurate placement of events on a timeline, to the resolution possible by the number of time expressions in the document. We suspect that this finer grained information will also be more useful for downstream applications like coreference, for which coarse information was found to be useful. The approach we develop is a supervised machine

Proceedings of the 2013 Workshop on Biomedical Natural Language Processing (BioNLP 2013), pages 18–26,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

learning approach in which pairs of time expressions and events are classified as CONTAINS or not. The specific approach is a support vector machine using both standard feature kernels and tree kernels, a novel approach to this problem in this domain that has shown promise on other relation extraction tasks.

This work makes use of a new corpus we developed as part of the THYME[1] project (Temporal History of Your Medical Events) focusing on temporal events and relations in clinical text. This corpus consists of clinical and pathology notes on colorectal cancer from Mayo Clinic. Gold standard annotations include Penn Treebank-style phrase structure in addition to clinically relevant temporal annotations like clinical events, temporal expressions, and various temporal relations.

2 Background and Related Work

2.1 Annotation Methodology

The THYME annotation guidelines[2] detail the extension of TimeML (Pustejovsky et al., 2003b) to the annotations of events, temporal expressions and temporal relations in the clinical domain. In summary, an EVENT is anything that is relevant to the clinical timeline. Temporal expressions (TIMEX3s) in the clinical domain are similar to those in the general domain with two exceptions. First, TimeML sets and frequencies occur much more often in the clinical domain, especially with regard to medications and treatments (*Claritin 30mg **twice daily***). The second deviation is a new type of TIMEX3 – PREPOSTEXP which covers temporally complex terms like *preoperative*, *postoperative*, and *intraoperative*.

EVENTs and TIMEX3s are ordered on a timeline through temporal TLINKs which range from fairly coarse (the relation to document time creation) to fairly granular (the explicit pairwise TLINKs between EVENTs and/or TIMEX3s). Of note for this work, the CONTAINS relation between a TIMEX3 and an EVENT means that the span of the EVENT is completely within the span of the TIMEX3. The interannotator agreement F1-score for CONTAINS for the set of documents used here was 0.60.

2.2 Narrative Containers

One relatively new concept for marking temporal relations is that of narrative containers, as in Pustejovsky and Stubbs (2011). Narrative containers are time spans which are central to the discourse and often subsume multiple events and time expressions. They are often anchored by a time expression, though more abstract events may also act as anchors. Using the narrative container framework significantly reduces the number of explicit TLINK annotations yet retains a relevant degree of granularity enabling inferencing.

Consider the following clinical text example with DocTime of February 8.

> *The patient recovered well after her initial first surgery on December 16th to remove the adenocarcinoma, although on the evening of January 3rd she was admitted with a fever and treated with antibiotics.*

There are three narrative containers in this snippet – (1) the broad period leading up to the document creation time which includes the events of *recovered* and *adenocarcinoma*, (2) *December 16th*, which includes the events of *surgery* and *remove*, and (3) *January 3rd*, which includes the events of *admitted*, *fever*, and *treated*.

Using only the relation to the document creation time would provide too coarse of a timeline resulting in collapsing the three narrative containers (the coarse time bins of Raghavan et al. (2012) would collapse all events into the *before admission* category). On the other hand, marking explicit links between every pair of events and temporal expressions would be tedious and redundant. In this example, there is no need to explicitly mark that, for instance, *fever* was AFTER *surgery*, because we know that the fever happened on January 3rd and that the surgery happened on December 16th, and that January 3rd is AFTER December 16th. With the grouping of EVENTs in this way, we can infer the links between them and reduce annotator effort. Narrative containers strike the right balance between parsimony and expressiveness.

2.3 Related Work

Of course, the possibility of annotating temporal containment relations was allowed by even the earliest versions of the TimeML specification using TLINKs with the relation type INCLUDES. However, TimeML is a specification not a guideline, and as such, the way in which temporal relations have been annotated has varied widely and no

[1] http://clear.colorado.edu/TemporalWiki

[2] Annotation guidelines are posted on the THYME wiki.

corpus has previously been annotated with narrative containers in mind. In the TimeBank corpus (Pustejovsky et al., 2003a), annotators annotated only a sparse, mostly disconnected graph of the temporal relations that seemed salient to them. In TempEval 2007 and 2010 (Verhagen et al., 2007; Verhagen et al., 2010), annotators annotated only relations in specific constructions – e.g. all pairs of events and times in a sentence – and used a restricted set of relation types that excluded the INCLUDES relation. TempEval 2013 (UzZaman et al., 2013) allowed INCLUDES relations, but again only in particular constructions or when the relation seemed salient to the annotators. The 2012 i2b2 Challenge[3], which provided TimeML annotations on clinical data, annotated the INCLUDES relation, but merged it with other relations for the evaluation due to low inter-annotator agreement.

Since no narrative container-annotated corpora exist, there are also no existing models for extracting narrative container relations. However, we can draw on the various methods applied to related temporal relation tasks. Most relevant is the work on linking events to timestamps. This was one of the subtasks in TempEval 2007 and 2010, and systems used a variety of features including words, part-of-speech tags, and the syntactic path between the event and the time (Bethard and Martin, 2007; Llorens et al., 2010). Syntactic path features were also used in the 2012 i2b2 Challenge, where they provided gains especially for intra-sentential temporal links (Xu et al., 2013).

Recent research has also looked to syntactic tree kernels for temporal relation extraction. Mirroshandel et al. (2009) used a path-enclosed tree (i.e., selecting only the sub-tree containing the event and time), and used various weighting scheme variants of this approach on the TimeBank (Pustejovsky et al., 2003a) and Opinion[4] corpora. Hovy et al. (2012) used a flat tree structure for each event-time pair, including only token-based information (words, part of speech tags) between the event and time, and found that adding such tree kernels on top of a baseline set of features improved event-time linking performance on the TempEval 2007 and Machine Reading corpora (Strassel et al., 2010). While Mirroshandel et al. saw improvements using a representation with syntactic structure, Hovy et al. used the flat tree structure because they found that "using a full-parse syntactic tree as input representation did not help performance." Thus, it remains an open question exactly where and when syntactic tree kernels will help temporal relation extraction.

3 Methods

Inspired by this prior work, we treat the narrative container extraction task as a within-sentence relation extraction task between time and event mentions. For each sentence, this approach iterates over every gold standard annotated EVENT, pairing it with each TIMEX3 in the sentence, and uses a supervised machine learning algorithm to classify each pair as related by the CONTAINS relation or not. Training examples are generated in the same way, with pairs corresponding to annotated links marked as positive examples and all others marked as negative. We investigate a variety of features for the classifier as well as a variety of tree kernel combinations.

This straightforward approach does not address all relation pairs, setting aside event-event relations and inter-sentential relations, which are both likely to require different approaches.

3.1 SVM with Tree Kernels

The machine learning approach we use is support vector machine (SVM) with standard feature kernels, tree kernels, and composite kernels that combine the two. SVMs are used extensively for classification tasks in natural language processing, due to robust performance and widely available software packages. We take advantage of the ability in SVMs to represent structured features such as trees using *convolution kernels* (Collins and Duffy, 2001), also known as *tree kernels*. This kernel computes similarity between two tree structures by computing the number of common sub-trees, with a weight parameter to discount the influence of larger structural similarities. The specific formalism we use is sometimes called a *subset tree* kernel (Moschitti, 2006), which checks for similarity on subtrees of all sizes, as long as each subtree has its production rule completely expanded.

A useful property of kernels is that a linear combination of two kernels is guaranteed to be a kernel (Cristianini and Shawe-Taylor, 2000). In addition, the product of two kernels is also a kernel. This means that it is simple to combine traditional feature-based kernels used in SVMs (linear,

[3] http://i2b2.org/NLP/TemporalRelations

[4] Also known as the AQUAINT TimeML corpus – http://www.timeml.org

polynomial, radial basis function) with tree kernels representing structural information. This approach of using *composite kernels* has been widely used in the task of relation extraction where syntactic information is presumed to be useful, but is hard to represent as traditional numeric features.

We investigate a few different composite kernels here, including a linear combination:

$$K_C(o_1, o_2) = \tau * K_T(t_1, t_2) + K_F(f_1, f_2) \quad (1)$$

where a composite kernel K_C operates on objects o_j composed of features f_j and tree t_j, by adding a tree kernel K_T weighted by τ to a feature kernel K_F. We also use a composite kernel that takes the product of kernels:

$$K_C(o_1, o_2) = K_T(t_1, t_2) * K_F(f_1, f_2) \quad (2)$$

Sometimes it is beneficial to make use of multiple syntactic "views" of the same instance. Below we will describe many different tree representations, and the tree kernel framework allows them to all be used simultaneously, by simply summing the similarities of the different representations and taking the combined sum as the tree kernel value:

$$K_T(\{t_1^1, t_1^2 \ldots, t_1^N\}, \{t_2^1, t_2^2, \ldots, t_2^N\}) = \sum_{i=1}^{N} K_T(t_1^i, t_2^i) \quad (3)$$

where i indexes the N different tree views. In all kernel combinations we compute the normalized version of both the feature and tree kernels so that they can be combined on an even footing.

The actual implementations we use for training are the SVM-LIGHT-TK package (Moschitti, 2006), which is a tree kernel extension to SVM^{light} (Joachims, 1999). At test time, we use the SVM-LIGHT-TK bindings of the ClearTK toolkit (Ogren et al., 2009) in a module built on top of Apache cTAKES (Savova et al., 2010), to take advantage of the pre-processing stages.

3.2 Flat Features

The flat features developed for the standard feature kernel include the text of each argument as a whole, the tokens of each argument represented as a bag of words, the first and last word of each argument, and the preceding and following words of each argument as bags of words. The token context between arguments is also represented using the text span as a whole, the first and last words, the set of words represented as a bag of words, and the distance between the arguments. In addition, part of speech (POS) tag features are extracted for each mention, with separate bag of POS tag features for each argument. The POS features are generated by the cTAKES POS tagger.

We also include semantic features of each argument. For event mentions, we include a feature marking the *contextual modality*, which can take on the possible values *Actual*, *Hedged*, *Hypothetical*, or *Generic*, which is part of the gold standard annotations. This feature was included as it was presumed that actual events are more likely to have definite time spans, and thus be related to times, than hypothetical or generic mentions of events. For time mentions we include a feature for the *time class*, with possible values of *Date*, *Time*, *Duration*, *Quantifier*, *Set*, or *Prepostexp*. The time class feature was used as it was hypothesized that dates and times are more likely to contain events than sets (e.g., *once a month*).

3.3 Tree Kernel Representations

We leverage existing tree kernel representations for this work, using some directly and others as starting point to a domain-specific representation.

First, we take advantage of the (relatively) flat structured tree kernel representations of Hovy et al. (2012). This representation uses lexical items such as POS tags rather than constituent structure, but places them into an ordered tree structure, which allows tree kernels to use them as a bag of items while also taking advantage of ordering structure when it is useful. Figure 1 shows an example tree for an event-time pair for which a relation exists, where the lexical information used is POS tag information for each term (the representation that Hovy et al. found most useful). We also used a version of this representation where the surface form is used instead of the POS tag.

While Hovy et al. showed positive results using this representation over just standard features, it is still somewhat constrained in its ability to represent long distance relations. This is because the subset tree kernel compares only complete rule productions, and with long distance relations a flat tree structure will have a production that is too big to learn. Alternatively, tree kernel representations can be based on constituent structure, as is common in the relation extraction literature. This will

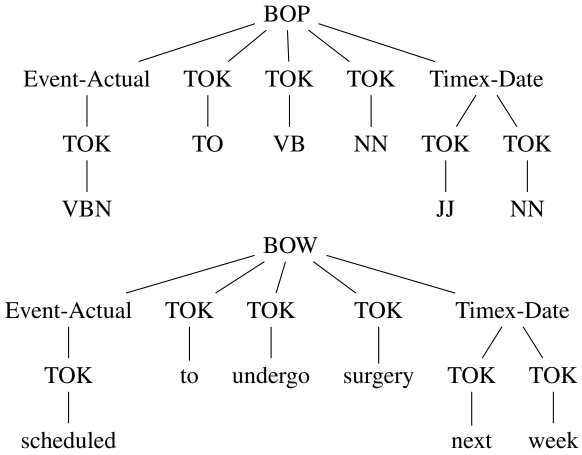

Figure 1: Two trees indicating the flat tree kernel representation. Above is the bag of POS tags version; below is the bag of words version.

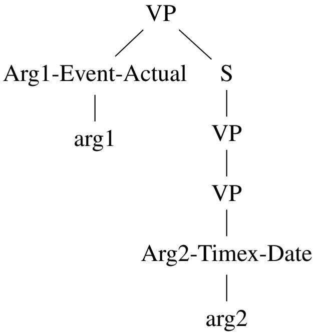

Figure 2: Path Tree (PT) representation

hopefully allow for the representation of longer distance relations by taking advantage of syntactic sub-structure with smaller productions. The representations used here are known as Feature Trees (FT), Path Trees (PT) and Path-Enclosed Trees (PET).

The Feature Tree representation takes the entire syntax tree for the sentence containing both arguments and inserts semantic information about those arguments. That information includes the argument type (EVENT or TIMEX) as an additional tree node above the constituent enclosing the argument. We also append semantic class information to the argument (contextual modality for events, time class for times), as in the flat features.

The Feature Tree representation is not commonly used, as it includes an entire sentence around the arguments of interest, and that may include a great deal of unrelated structure that adds noise to the classifier. Here we include it in an attempt to get to the root of an apparent discrepancy in the tree kernel literature, as explained in Section 2, in which Hovy et al. (2012) report a negative result and Mirroshandel et al. (2009) report a positive result for using constituency structure in tree kernels for temporal relation extraction.

The Path Tree representation uses a sub-tree of the whole constituent tree, but removes all nodes that are not along the path between the two arguments. Path information has been used in standard feature kernels (Pradhan et al., 2008), with each individual path being a possible boolean feature. Another representation making use of the path tree takes contiguous subsections of the path tree, or "path n-grams," in an attempt to combat the sparsity of using the whole path (Zheng et al., 2012). By using the path representation with a tree kernel, the model should get the benefit of all different sizes of path n-grams, up to the size of the whole path. This representation is augmented by adding in argument nodes with event and time features, as in the Feature Tree. Unlike the Feature Tree and the PET below, the Path Tree representation does not include word nodes, because the important aspect of this representation is the labels of the nodes on the path between arguments. Figure 2 shows an example of what this representation looks like.

The Path-Enclosed Tree representation is based on the smallest sub-tree that encloses the two proposed arguments. This is a representation that has shown value in other work using tree kernels for relation extraction (Zhang et al., 2006; Mirroshandel et al., 2009). The information contained in the PET representation is a superset of that contained in the Path Tree representation, since it includes the full path between arguments as well as the structure between arguments and the argument text. This means that it can take into account path information while also considering constituent structure between arguments that may play a role in determining whether the two arguments are related. For example, temporal cue words like *after* or *during* may occur between arguments and will not be captured by Path Trees. Like the PT representation, the PET representation is augmented with the semantic information specified above in the Feature Tree representation. Figure 3 shows an example of this representation.

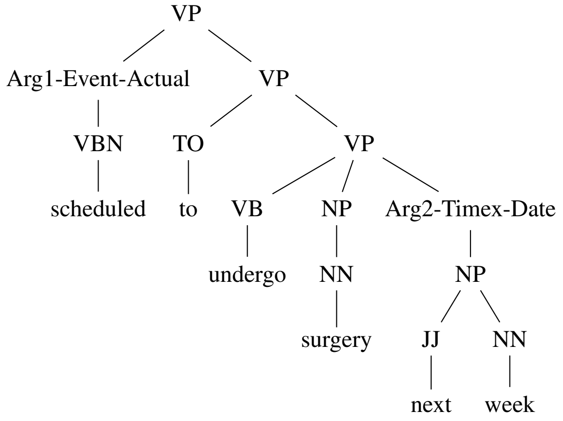

Figure 3: Path-Enclosed Tree representation

4 Evaluation

The corpus we used for evaluations was described in Section 2. There are 78 total notes in the corpus, with three notes for each of 26 patients. The data is split into training (50%), development (25%), and test (25%) sections based on patient number, so that each patient's notes are all in the same section. The combined training and development set used for final training consists of 4378 sentences with 49,050 tokens, and 7372 events, 849 time expressions, and 2287 CONTAINS relations. There were 774 positive instances of CONTAINS in the training data, with 1513 negative instances. For constituent structure and features we use the gold standard treebank and event and time features from our corpus. Preliminary work suggests that automatic parses from cTAKES do not harm performance very much, but the focus of this work is on the relation extraction so we use gold standard parses. All preliminary experiments were done using the development set for testing.

We designed a set of experiments to examine several hypotheses regarding extraction of the CONTAINS relation and the efficacy of different tree kernel representations. The first two configurations test simple rule-based baseline systems, CLOSEST-P and CLOSEST-R, for distance-related decision rule systems meant to optimize precision and recall, respectively. CLOSEST-P hypothesizes a CONTAINS link between every TIMEX3 and the closest annotated EVENT, which will make few links overall. CLOSEST-R hypothesizes a CONTAINS link between every EVENT and the closest TIMEX3, which will make many more links.

The next configuration, *Flat Features*, uses the token and part of speech features along with argument semantics features, as described in Section 3. While this feature set may not seem exhaustive, in preliminary work many traditional relation extraction features were tried and found to not have much effect. This particular configuration was tested because it is most comparable to the bag of word and bag of POS kernels from Hovy et al. (2012), and should help show whether the tree kernel is providing anything over an equivalent set of basic features.

We then examine several composite kernels, all using the same feature kernel, but using different tree kernel-based representations. First, we use a composite kernel which uses the bag of word and bag of POS tree views, as in Hovy et al. (2012). Next, we add in two additional tree views to the tree kernel, Path-Enclosed Tree and Path Tree, which are intended to examine the effect of using traditional syntax, and the long distance features that they enable. The final experimental configuration replaces the PET and PT representations from the last configuration with the Feature Tree representation. This tests the hypothesis that the difference between positive results for tree kernels in this task (as in, say, Mirroshandel et al. (2009)) and negative results reported by Hovy et al. (2012) is the difference between using a full-parse tree and using standard sub-tree representations.

For the rule-based systems, there are no parameters to tune. Our machine-learning systems are based on support vector machines (SVM), which require tuning of several parameters, including kernel type (linear, polynomial, and radial basis function), the parameters for each kernel, and c, the cost of misclassification. Tree kernels introduce an additional parameter λ for weighting large structures, and the use of a composite kernel introduces parameters for which kernel combination operator to use, and how to weight the different kernels for the sum operator.

For each machine learning configuration, we performed a large grid search over the combined parameter space, where we trained on the training set and tested on the development set. For the final experiments, the parameters were chosen that optimized the F1 score on the development set. Qualitatively, the parameter tuning strongly favored configurations which combined the kernels using the sum operator, and recall and precision were strongly correlated with the SVM parameter c. Using these parameters, we then trained on the combined training and development sets and tested on the official test set.

4.1 Evaluation Metrics

The state of evaluating temporal relations has been evolving over the past decade. This is partially due to the inferential properties of temporal relations, because it is possible to define the same set of relations using different set of axioms. To take a very simple example, given a gold set of relations A<B and B<C, and given the system output A<B, A<C and B<C, if one were to compute a plain precision/recall metric, then the axiom A<C would be counted against the system, when one can easily infer from the gold set of relations that it is indeed correct. With more relations the inference process becomes more complex.

Recently there has been some work trying to address the shortcomings of the plain F1 score (Muller and Tannier, 2004; Setzer et al., 2006; UzZaman and Allen, 2011; Tannier and Muller, 2008; Tannier and Muller, 2011). However, the community has not yet come to a consensus on the best evaluation approach. Two recent evaluations, TempEval-3 (UzZaman et al., 2013) and the 2012 i2b2 Challenge (Sun et al., 2013), used an implementation of the proposal by (UzZaman and Allen, 2011). However, as described in Cherry et al. (2013), this algorithm, which uses a greedy graph minimization approach, is sensitive to the order in which the temporal relations are presented to the scorer. In addition, the scorer is not able to give credit for non-redundant, non-minimum links (Cherry et al., 2013) as with the the case of the relation A<C mentioned earlier.

Considering that the measures for evaluating temporal relations are still evolving, we decided to use plain F-score, with recall and precision scores also reported. This score is computed across all intra-sentential EVENT-TIMEX3 pairs in the gold standard, where precision = $\frac{\#\ correct\ predictions}{\#\ predictions}$, recall = $\frac{\#\ correct\ predictions}{\#\ gold\ standard\ relations}$, and F1 score = $\frac{2*precision*recall}{precision+recall}$.

4.2 Experimental Results

System	Precision	Recall	F1
CLOSEST-P	0.754	0.537	0.627
CLOSEST-R	0.502	**0.947**	0.656
Flat Features (FF)	0.705	0.593	0.645
FF+Bag Trees (BT)	0.649	0.728	0.686
FF+BT+PET+PT	**0.770**	0.707	**0.737**
FF+BT+FT	0.691	0.691	0.691

Table 1: Table of results of main experiments.

Results are shown in Table 1. Rule-based baselines perform reasonably well, but are heavily biased in terms of precision or recall. The machine learning baseline cannot even obtain the same performance as the CLOSEST-R rule-based system, though it is more balanced in terms of precision and recall. Using a composite kernel which adds in the flat token-based tree kernels improves performance over the standard feature kernel by 4.1 points. Adding in the Path Tree and Path-Enclosed Tree constituency-based trees along with the flat trees improves F1 score to our best result of 73.7. Finally, replacing PT and PET representations with the Feature Tree representation does not offer any performance improvement over the Flat Features + Bag Trees configuration.

4.3 Error Analysis

We performed error analysis on the outputs of the best-performing system (FF+BT+PET+PT in Table 1). First, we note that the parameter search was optimized for F1. This resulted in the highest-scoring configuration using a composite kernel with the sum operator, polynomial kernel for the secondary kernel, $\lambda = 0.5$, tree kernel weight (T) of 0.1, and $c = 10.0$. This high value of c and low value of T results in higher precision and lower recall, but there were configurations with lower c and higher T which made the opposite tradeoff, with only marginally worse F1-score. For the purposes of error analysis, however, this configuration leads to a focus on false negatives.

First, the false positives contained many relations that were legitimately ambiguous or possible annotator errors. An example ambiguous case is *She is currently being treated on the Surgical Service for...*, in which the system generates the relation CONTAINS(*currently*, *treated*), but the gold standard labels as OVERLAP. This example is ambiguous because it is not clear from just the linguistic context whether the treatment is wholly contained in the small time window denoted by *currently*, or whether it started a while ago or will continue into the future. There are many similar cases where the event is a disease/disorder type, and the specific nature of the disease is important to understanding whether this is a CONTAINS or OVERLAP relation, specifically understanding whether the disease is chronic or more acute.

Another source of false positives were where the event and time were clearly related, but not with CONTAINS. In the example *reports that she has been having intermittent bleeding since May of 1998*, the term *since* clearly indicates that this is a BEGINS-ON relation between *bleeding* and *May of 1998*. This is a case where having other temporal relation classifiers may be useful, as they can compete and the relation can be assigned to whichever classifier is more confident.

False negatives frequently occurred in contexts where the event and time were far apart. Syntactic tree kernels were introduced to help improve recall on longer-distance relations, and were successful up to a limit. However, certain examples are so far apart that the algorithm may have had difficulty sorting noise from important structure. For example, the system did not find the CONTAINS(*October 27, 2010*, *oophorectomy*) relation in the sentence:

> *October 27, 2010, Dr. XXX performed exploratory laparotomy with an transverse colectomy and Dr. YYY performed a total abdominal hysterectomy with a bilateral salpingo-oophorectomy.*

Here, while the date may be part of the same sentence as the event, the syntactic relation between the pair is not what makes the relation; the date is acting as a kind of discourse marker that indicates that the following events are contained. This suggests that discourse-level features may be useful even for the intra-sentential classification task.

Other false negatives occurred where there was syntactic complexity, even on shorter examples. The subset tree kernel used here matches complete rule productions, and across complex structure with large productions, the chances of finding similarity decreases substantially. Thus, events within coordination or separated from the time by clause breaks are more difficult to relate to the time due to the multiple different ways of relating these different syntactic elements.

Finally, there are some examples where the anchor of a narrative container is an event with multiple sub-events. In these cases, the system performs well at relating a time expression to the anchor event, but may miss the sub-events that are farther away. This is a case where having an event-event TLINK classifier, then applying deterministic closure rules, would allow a combined system to link the sub-events to the time expression.

5 Discussion and Conclusion

In this paper we have developed a system for automatically identifying CONTAINS relations in clinical text. The experiments show first that a machine learning approach that intelligently integrates constituency information can greatly improve performance over rule-based baselines. We also show that the tree kernel approach, which can model sequence better than a bag of tokens-style approach, is beneficial even when it uses the same features. Finally, the experiments show that choosing the correct representation is important for tree kernel approaches, and specifically that using a full parse tree may give inferior performance compared to sub-trees focused on the structure of interest.

In general, there is much work to be done in the area of representing temporal information in clinical records. Many of the inputs to the algorithm described in this paper need to be extracted automatically, including time expressions and events. Work on relations will focus on adding features to represent discourse information and richer representation of event semantics. Discourse information may help with the longer-distance errors, where the time expression acts almost as a topic for an extended description of events. Better understanding of event semantics, such as whether a disease is chronic or acute, or typical duration for a treatment, may help constrain relations. In addition, we will explore the effectiveness of using dependency tree structure, which has been useful in the domain of extracting relations from the biomedical literature (Tikk et al., 2013).

Acknowledgements

The work described was supported by Temporal History of Your Medical Events (THYME) NLM R01LM010090 and Integrating Informatics and Biology to the Bedside (i2b2) NCBO U54LM008748. Thanks to the anonymous reviewers for thorough and insightful comments.

References

Naushad UzZaman, Hector Llorens, et al. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM),* *Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Ashwin N Ananthakrishnan, Tianxi Cai, et al. 2013. Improving case definition of Crohn's disease and ulcerative colitis in electronic medical records using natural language processing: a novel informatics approach. *Inflammatory bowel diseases*.

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 Semantic Evaluations (SemEval-2007)*, pages 129–132.

Colin Cherry, Xiaodan Zhu, et al. 2013. A la recherche du temps perdu: extracting temporal relations from medical text in the 2012 i2b2 NLP challenge. *Journal of the American Medical Informatics Association*, March.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Neural Information Processing Systems*.

Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Dirk Hovy, James Fan, et al. 2012. When did that happen?: linking events and relations to timestamps. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 185–193. Association for Computational Linguistics.

Thorsten Joachims. 1999. Making large scale svm learning practical. In B. Schlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. Universität Dortmund.

Chen Lin, Helena Canhao, et al. 2012. Feature engineering and selection for rheumatoid arthritis disease activity classification using electronic medical records. In *Proceedings of ICML Workshop on Machine Learning for Clinical Data*.

Hector Llorens, Estela Saquete, and Borja Navarro. 2010. TIPSem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291. Association for Computational Linguistics.

Seyed Abolghasem Mirroshandel, M Khayyamian, and GR Ghassem-Sani. 2009. Using tree kernels for classifying temporal relations between events. *Proc. of the PACLIC23*, pages 355–364.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Machine Learning: ECML 2006*, pages 318–329. Springer.

Philippe Muller and Xavier Tannier. 2004. Annotating and measuring temporal relations in texts. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Philip V. Ogren, Philipp G. Wetzler, and Steven J. Bethard. 2009. ClearTK: a framework for statistical natural language processing. In *Unstructured Information Management Architecture Workshop at the Conference of the German Society for Computational Linguistics and Language Technology*, 9.

Sameer S Pradhan, Wayne Ward, and James H Martin. 2008. Towards robust semantic role labeling. *Computational Linguistics*, 34(2):289–310.

James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160.

James Pustejovsky, Patrick Hanks, et al. 2003a. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40.

James Pustejovsky, José Casta no, et al. 2003b. Timeml: Robust specification of event and temporal expressions in text. In *Fifth International Workshop on Computational Semantics (IWCS-5)*.

Preethi Raghavan, Eric Fosler-Lussier, and Albert M Lai. 2012. Temporal classification of medical events. In *Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pages 29–37. Association for Computational Linguistics.

Guergana K. Savova, James J. Masanz, et al. 2010. Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. *J Am Med Inform Assoc*, 17(5):507–513.

Andrea Setzer, Robert Gaizauskas, and Mark Hepple. 2006. The role of inference in the temporal annotation and analysis of text. *Language Resources and Evaluation*, 39(2-3):243–265, February.

Stephanie Strassel, Dan Adams, et al. 2010. The DARPA machine reading program - encouraging linguistic and reasoning research with a series of reading tasks. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, April.

Xavier Tannier and Philippe Muller. 2008. Evaluation metrics for automatic temporal annotation of texts. *Proceedings of the Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*.

Xavier Tannier and Philippe Muller. 2011. Evaluating temporal graphs built from texts via transitive reduction. *J. Artif. Int. Res.*, 40(1):375413, January.

Domonkos Tikk, Illés Solt, et al. 2013. A detailed error analysis of 13 kernel methods for protein-protein interaction extraction. *BMC bioinformatics*, 14(1):12.

Naushad UzZaman and James Allen. 2011. Temporal evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, page 351–356.

Marc Verhagen, Robert Gaizauskas, et al. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80.

Marc Verhagen, Roser Sauri, et al. 2010. Semeval-2010 task 13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden, July. Association for Computational Linguistics.

RA Wilke, H Xu, et al. 2011. The emerging role of electronic medical records in pharmacogenomics. *Clinical Pharmacology & Therapeutics*, 89(3):379–386.

Yan Xu, Yining Wang, et al. 2013. An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association : JAMIA*.

Min Zhang, Jie Zhang, and Jian Su. 2006. Exploring syntactic features for relation extraction using a convolution tree kernel. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 288–295.

Jiaping Zheng, Wendy W Chapman, et al. 2012. A system for coreference resolution for the clinical narrative. *Journal of the American Medical Informatics Association*, 19:660–667.