Published in final edited form as:
J Biomed Inform. 2013 December ; 46(0): . doi:10.1016/j.jbi.2013.08.003.

Classifying Temporal Relations in Clinical Data: A Hybrid, Knowledge-Rich Approach

Jennifer D'Souza[a] and **Vincent Ng**[b]
[a,b]Human Language Technology Research Institute, University of Texas at Dallas, Richardson, TX 75080, USA

Abstract

We address the TLINK track of the 2012 i2b2 challenge on temporal relations. Unlike other approaches to this task, we (1) employ sophisticated linguistic knowledge derived from semantic and discourse relations, rather than focus on morpho-syntactic knowledge; and (2) leverage a novel combination of rule-based and learning-based approaches, rather than rely solely on one or the other. Experiments show that our knowledge-rich, hybrid approach yields an F-score of 69.3, which is the best result reported to date on this dataset.

Keywords

temporal relations; semantic relations; discourse relations; hybrid approaches

1. Introduction

Temporal relation extraction and classification, one of the most important temporal information extraction (IE) tasks, involves extracting *entities* (i.e., events and time expressions) from a text document and determining their temporal relations with each other. The creation of the TimeBank corpus [16], as well as the organization of the TempEval-1 [21] and TempEval-2 [22] evaluation exercises, have facilitated the development and evaluation of temporal relation classification systems for the new domain.

More recently, the 2012 i2b2 Challenge has focused on tasks related to extracting and classifying temporal relations from clinical data that comprised patient discharge reports [17]. Each report is composed of two sections, *history of present illness* and *hospital course*. The shared task was subdivided into 3 tracks: 1. EVENT/TIMEX3 extraction; 2. TLINK extraction; and 3. End-to-end system. Our goal in this paper is to advance the state of the art on the TLINK extraction track.

The TLINK extraction track is composed of two tasks. Both tasks assume as input a document manually annotated with *entities*, which are either events or time expressions, as mentioned above. The first task, EVENT/TIMEX3-SCT TLINK Type Prediction, is a three-class classification task: given an event *e* and the creation time *t* of one of the two aforementioned sections of a report, determine whether *e* and *t* have a **Before**, **After**, or **Overlap** relation. The second task, EVENT-EVENT/EVENT-TIMEX3 TLINK Type

Prediction, is a four-class classification task: given an event-event or event-time pair, determine whether the two elements of the pair have a **Before**, **After**, or **Overlap** relation, or whether no temporal relation exists between them. While these two tasks can in principle be tackled in any order, in our approach we will use the output of the first task when addressing the second task, effectively yielding a pipeline architecture.

Our approach to TLINK extraction can be distinguished from other approaches, including those developed for the news domain and the clinical domain, in two respects. The first involves a large-scale expansion of the linguistic features made available to the classification system. Recall that existing approaches have typically relied on morpho-syntactic features, as well as a few semantic features extracted from WordNet synsets and VerbOcean's [4] semantic relations. In contrast, we propose not only novel lexical and grammatical features, but also sophisticated features involving semantics and discourse. Most notably, we propose (1) semantic features encoding PropBank-style predicate-argument relations, and (2) discourse features encoding Penn Discourse TreeBank (PDTB) style [14] discourse relations.

Second, while the vast majority of approaches to temporal relation classification adopted in the shared task are learning-based, we propose a system architecture in which we combine a learning-based approach and a rule-based approach. Our motivation behind adopting a hybrid approach stems from two hypotheses. First, a rule-based method could better handle the skewed class distributions that exist in the dataset for the EVENT-EVENT/EVENT-TIMEX3 TLINK Type Prediction task. Second, better decision rules could be formed by leveraging human insights to combine the available linguistic features than by using fully automatic machine learning methods. Note that while rule-based approaches have been explored for this task and shown to underperform learning-based approaches [11], to our knowledge they have not been used in combination with learning-based approaches. Moreover, while the rules employed in previous work are created based on intuition [e.g., 11,15], our rules are created in a data-driven manner via a manual inspection of the annotated temporal relations in the i2b2 corpus.

Another unique feature of our approach concerns the way we apply machine learning to temporal relation classification. While in existing learning-based approaches to this task typically only one classifier is trained to determine the temporal relation between two entities, in our approach we train multiple classifiers, each of which is specialized in classifying a different type of entity pair.

Experiments on the i2b2 Clinical Temporal Relations Challenge corpus demonstrate the effectiveness of our knowledge-rich, hybrid approach to temporal relation classification: we achieved an F-score of 69.3% on the test set, which is the best result reported to date on this dataset.

The rest of the paper is organized as follows. Section 2 provides a brief overview of related work on temporal relation classification. Section 3 describes our methods, including a brief overview of the i2b2 corpus, our evaluation methodology and our approaches to the two tasks in the TLINK extraction track. We present experimental results in Section 4 and conclude in Section 5.

2. Related Work

A number of corpora have been developed over the years for evaluating temporal relation extraction and classification systems, including TimeBank [16], those used for the TempEval-2007 [21] and TempEval-2010 [22] evaluation exercises, and most recently the

i2b2 corpus. Below we organize existing approaches to this task roughly based on the corpus used in their evaluation.

2.1. The TimeBank Corpus

Early approaches to temporal relation extraction and classification, such as Mani et al. [11], Chambers et al. [1], and Chambers and Jurafsky [2], were evaluated on the TimeBank corpus [16], which consists of 183 newswire articles manually annotated with the events, times, and their temporal relations. 14 relation types are defined and used to annotate the temporal relations in this corpus.

Mani et al. [11] approached the temporal relation classification task as a 6-class classification task reduced from the overall 14 classes. They noted that not all temporal relations were annotated in the corpus. As an example, consider a document where A and B are annotated as having a **Before** relation, and B and C are also annotated as having a **Before** relation. Intuitively, although A and C should also have a **Before** relation, they are not annotated. Mani et al. [11] attributed such omissions to annotator fatigue, and note that such omissions translate to the creation of a smaller number of training instances. To address this problem, they applied transitive closure to the existing annotations to obtain additional Tlink annotations. They trained a maximum entropy classifier for event-event and event-time classification. For classifying event-event relations they employed features computed based on event attributes, event string, preposition and context indicating if the events have the same tense and same aspect. For classifying event-time relations, a similar set of features is computed based on the attributes of the time expression in the pair. They also proposed a rule-based system with rules derived from human intuition and showed that the machine learning system outperforms the rule-based system.

Chambers et al. [1] also employed a machine learning approach to event-event temporal relation classification. In addition to the features used in Mani et al.'s work, they employed features such as event attributes, the part of speech tags of the event and tokens in context, syntactic dominance relation from the parse tree, information on whether the event is contained in a prepositional phrase. They had an additional feature that splits the data based on whether the two events in an event pair appear in the same sentence or not. They approached the task as a 6-way classification task out of the 14 temporal relations, though these 6 relations are not identical to the ones used by Mani et al. [11].

As discussed above, both Mani et al. [11] and Chambers et al. [1] trained a pairwise classifier for classifying the temporal relation of a given pair of events. Chambers and Jurafsky [2] noted that one of the weaknesses of such a pairwise approach is that global constraints cannot be enforced. For instance, it is possible for the pairwise classifier to determine that A occurs **Before** B and B occurs **Before** C, and that A occurs **After** C. In other words, since each event pair is classified independently of the others, global constraints such as transitivity constraints cannot be enforced. To address this problem, Chambers and Jurafsky [2] proposed a formulation based on integer linear programming to enforce global constraints.

2.2. The TempEval-2007 Corpus

The TempEval-2007 [21] shared task marks the beginning of wider community initiative towards temporal relation classification. The classification tasks defined were mainly for 3 classes **Before**, **After**, and **Overlap** with 3 additional sparse disjunctive classes namely **Before-Or-Overlap**, **Overlap-Or-After** and **Vague**. The tasks related to temporal identification were of three types: Task A involved classification of the temporal relation between an event and all temporal expressions within the same sentence; Task B involved

classification of the temporal relation between an event and the Document Creation Time (DCT); and Task C involved classification between main events in consecutive sentences. A new annotated corpus, which we refer to as the TempEval-2007 corpus, was created and used in the shared task evaluations.

Puscasu's [15] system achieved best performance for Tasks A and B. He inferred temporal relations from temporal reasoning applied on a temporally tagged parse tree formed from heuristic inferences based on semantic properties and syntactic types, etc. Min et al.'s [13] system achieved the best performance for Task C. They employed a machine learning approach using standard features such as entity string, head-word, attributes, context information, and the relation of a temporal expression with DCT.

2.3. The TempEval-2 Corpus

TempEval-2010 [22] was the second community-wide shared task organized in the area of temporal relation identification. Tasks A, B and C from TempEval-2007 [21] were arranged as Tasks C, D and E respectively. The additional task related to temporal classification was Task F, which required automatic classification of subordinated event relations within the same sentence (i.e., relations between two events where one event syntactically dominates the other). For certain tasks, annotated data may be available for multiple languages.

Lloren et al.'s TipSem system [10] performed well on all classification tasks for which Spanish data was available, which were tasks C and D. They formulated temporal relation classification as a sequential classification task, using features such as the heading preposition of the event or the time expression, the syntactic relation between the event and the time expression, the time position of the event with another related time expression, interval, the type of the time expression, and the temporal subordination role element associated with the event or the time expression. The TRIOS system performed the best on Task C and the TRIPS system performed the best on Task E. These two systems were created by the same team [20]. The unique aspect of their approach is in the use of Markov Logic Networks in inferring temporal relations. This framework accepted as features pre-written logical formulas on the basis of which it decides weights and reasons for a particular class. Another interesting aspect of their approach is that the two systems use system-generated events or time expressions along with their attributes as opposed to using gold attributes. Most of their event-related features that are commonly shared between Tasks C and E comprise information such as event attributes, event string, stem, and part-of-speech tags. The time-expression-related features used in Task C were the type, value, relation to the DCT. In Task E they formed formulas from the corresponding pairs of event attributes. In addition, they extracted feature information such as event constituent, event ontology type, and event lexical aspect which classifies the event as Event, State or Reporting.

The NCSU system [7] offered the highest performance on Task F. They formulated the problem as a supervised machine learning approach with Markov Logic. The unique aspect of their approach was their use of syntactic features such as governing verb and the part-of-speech tag of the governing verb, as well as lexical relations such as similarity, strength, and antonymy.

2.4. The i2b2 Corpus

The 2012 i2b2 challenge [17] marks a shift in the community-wide research initiative towards temporal relation extraction from newswire data to data from the clinical domain. Different from TempEval-2007 and TempEval-2010, systems built for temporal relation extraction in this challenge were required to both identify and classify the temporal relations. Three temporal relation types were considered, namely **Before**, **After**, and

Overlap. There were two main tracks in this challenge directly related to temporal relation classification: Track 2, also called the TLINK extraction track, provided text with gold annotations of event expressions and time expressions and required automatic annotations of temporal links, or TLINKs within the text; Track 3, also called the end-to-end system track, was essentially the same as track 2, except that automatically identified temporal entities are used when establishing the temporal links.

Cherry et al.'s [3] system was the best performing system in Track 2. Their overall feature group comprised lexical features including contextual n-grams, syntactic features from parse trees capturing part-of-speech and dependency information, semantic features including manually created categorized lexicons, and UMLS mappings through MetaMap.

Tang et al.'s [14] system achieved the best performance in Track 3. For classifying TLINKs between events and section creation times they used entity positional information, n-grams, part-of-speech tags, dependency relations, and entity attribute information. For classifying intra-sentence TLINKs, along with all the previous features, they used features such as the token distance between the two entities and a conjunction relation indicator between the paired entities. Additional features in their inter-sentence TLINK component captured semantic information, such as whether the events included the same positional words such as left and right, and whether the events had the same anatomic word (e.g., "arm" and "leg").

3. Methods

In this section, we present our approach to temporal relation classification. We begin by introducing the i2b2 corpus and our evaluation methodology (Section 3.1), followed by our methods for addressing the EVENT/TIMEX3-SCT TLINK Type Prediction task (Section 3.2) and the EVENT-EVENT/EVENT-TIMEX3 Type Prediction task (Section 3.3).

3.1. Corpus and Evaluation Methodology

Corpus: For evaluation, we use the 2012 i2b2 Clinical Temporal Relations Challenge corpus (henceforth the *i2b2 corpus*), which consists of 310 de-identified discharge summaries pre-partitioned into a training set (190 summaries) and a test set (120 summaries). In a summary, the *events*, the *time expressions*, the *temporal relation* between each event/time expression and the creation time of each of its two sections, as well as the *temporal relation* between the two elements of each event-event/event-time pair, are marked up. Note here that the temporal relations are marked up only in the training summaries, and the goal is to automatically extract and classify the temporal relations from the test summaries. An event, which can be a verb phrase, an adjective phrase, a noun phrase, or sometimes even an adverb that semantically refers to clinically relevant patient-related actions, contains various attributes, including the *type* of event[1], *polarity*, and *modality*. A time expression has a *type* attribute, which specifies whether it is a *date*, *time*, *duration*, or *frequency*, and its value is normalized based on TIMEX3 [16]. A temporal relation may order two events (as in sentence (1)), or it may anchor an event to a time expression (as in sentence (2)), where the time expression could be a section creation time:

(1) The patient was admitted for *treatment* of a presumed *aspiration pneumonia*.

(2) She had *a normal pancreas* at *that time*.

[1]Six types of events are defined, including TEST (e.g., *CT scan*), PROBLEM (e.g., *the tumor*), TREATMENT (e.g., operation), CLINICAL DEPARTMENTS (e.g., *ICU*), EVIDENTIAL information (e.g., *complained*), and clinically-relevant OCCURRENCE (e.g., *discharge*).

Each temporal relation has a *type*. For example, in (1), the event *treatment*, which has type TREATMENT, happens **After** event *aspiration pneumonia*, which has type PROBLEM and modality POSSIBLE; whereas in (2), the event *a normal pancreas*, which has type OCCURRENCE, has temporal **Overlap** relation with time expression *that time*, which has type DATE. A temporal relation is defined on an *ordered* pair: while the pair (*treatment, aspiration pneumonia*) has type **After**, the pair (*aspiration pneumonia, treatment*) has type **Before**.

Following the task definition, we assume that our temporal relation classification system is given gold (i.e., manually annotated) event and time expressions. The first task of the track aims to determine the temporal relation between an event/time expression and a section creation time. The second task aims to determine whether a temporal relation exists between the elements of an event-event or event-time pair; and if so, what its relation type is. While 12 relation types are defined and used to annotate the temporal relations in the i2b2 corpus, we follow the shared task definition and describe an approach that identifies only three broad relation types. Table 1 provides a brief description of these broad relation types and the relevant statistics.

Evaluation metrics: The precision, recall and F-measure scores reported in this paper are computed using the i2b2 shared task evaluation script. We use the default scoring scheme, where precision is defined as the total number of system output TLINKs that can be verified in the gold standard closure divided by the total number of system output TLINKs, and recall is the total number gold standard output TLINKs that can be verified in the system closure divided by the total number of gold standard output TLINKs. This metric serves as the standard for comparison between system outputs in this task.

3.2. Task 1: EVENT/TIMEX3-SCT TLINK Classification

In this subsection, we will describe our approach to the first task of the TLINK extraction track, which involves determining the temporal relation type between an event/time expression and a section creation time. We first describe how to recast the task as a sequence labeling task (Section 3.2.1) and then show the features used in the learning process (Section 3.2.2).

3.2.1. Classification as Sequence Labeling—Recall that each event/time expression in a patient discharge report (1) belongs to either the *history of present illness* section or the *hospital course* section; and (2) has a **Before**, **After**, or **Overlap** temporal relation with the creation time of its corresponding section. Hence, one simple way to approach the EVENT/TIMEX3-SCT TLINK task would be to classify each event/time expression as having a **Before**, **After**, or **Overlap** relation with the creation time of its corresponding section. Another way, however, would be to recast the task as a sequence labeling task, as described below, where each sequence corresponds to a sentence.

To employ sequence labeling, we label each token rather than each event/time expression in a given sentence with the relation type. Since we are labeling tokens, we follow the convention in sequence labeling and adopt the IOB labeling scheme, where we augment each relation type label X with B or I, where X ε{**Before**, **After**, **Overlap**}. Specifically, the label XB implies that the corresponding token *begins* an event/time expression that has the relation type X with its corresponding section creation time. Similarly, the label XI implies that the corresponding token is *inside* an event/time expression that has the relation type X with its corresponding section creation time. The label O should be used if the corresponding token corresponds to neither an event nor a time expression.

To train a model for assigning labels to tokens, we employ CRF++[2] as the sequence learner, creating one training instance for each token in each patient discharge report in the training set and deriving its class value (i.e., XB, XI, or O, where X is one of the three relation types) from the annotated data. Each instance represents the token under consideration and consists of the features described in Section 3.2.2.

There is a caveat, however. By recasting the task as a sequence labeling task, we simultaneously (1) identify (the boundaries of) event/time expressions and (2) determine the relation type between each expression with its section creation time. Hence, even though we assume as input gold event/time expressions, the trained CRF fails to exploit these gold boundaries. To address this problem, we augment the feature set in Section 3.2.2 with an additional binary feature that has the value 1 if and only if the corresponding token is part of an event/time expression. Even though there is still no guarantee that all event/time expressions will be identified by the CRF, the model should be able to learn that the class O appears if and only if the value of the additional feature is 0.

3.2.2. Features—To train the CRF, we represent each token with a set of features that are motivated by previous work on extracting gene names from biomedical literature [6,12,24] owing to the relevance of the word shape and syntactic features for the clinical dataset as well (see Table 2, where w_i denotes the current token).

Moreover, motivated by the TipSem Temporal Relation Categorization System [10], we also include features to provide the phrase-level information to the learner (see Table 3). Specifically, we employ four types of phrase-level features. The first type is Head Verb Phrase (VP). If the token under consideration is part of a VP, then the feature values are the verb heading the VP (verb_word$_h$) and its POS (verb_POS$_h$); otherwise, the values of both features are NULL. The second type is Governing VP. If the grandparent of the token under consideration is a VP, then the feature values are the verb heading this VP (verb-word$_g$) and its POS (verb_POS$_g$); otherwise, the values of both features are NULL. The third type is Governing preopositional phrase (PP). If the grandparent of the token under consideration is a PP, then the feature values is the preposition heading the PP; otherwise, the feature value is NULL. Finally, we have a set of token-based phrasal features. Specifically, if the token under consideration is part of an event, we employ features that encode the type, polarity, and modality of the embedding event; on the other hand, if the token is part of a time expression, we employ features that encode the type and modality of the embedding time expression. The example below serves to illustrate how all the phrase-level features values are identical for any given event/time expression.

(3) She has not received *prior radiation exposure*.

Structured Syntactic Parse: (S (NP (PRP She)) (VP (VBZ has) (RB not) (VP (VBN received) (PP (RB prior) (NP (NN radiation) (NN exposure))))) (..))

For the TREATMENT event *prior radiation exposure* in (3), from the sentence parse, each of the event tokens *prior*, *radiation*, and *exposure* appearing as consecutive lines in the CRF data file, receive the following phrase feature values:1) Head VP – NULL, 2) Governing VP – received, 3) Governing PP – NULL, and 4) Type_Polarity_Modality – TREATMENT_POS_FACTUAL.

[2]Available from http://crfpp.sourceforge.net.

3.3. Task 2: EVENT/TIMEX3-EVENT/TIMEX3 TLINK Classification

In this subsection, we describe our knowledge-rich, hybrid approach to the second task of the TLINK extraction track, which classifies each event-event and event-time pair in a patient discharge report as belonging one of four classes: **Before**, **After**, **Overlap**, and **No-Rel** (no temporal relation). To facilitate the evaluation of the contribution made by different components of our system, in the rest of this subsection we (1) describe the basic set of features used to train a relation type classifier (Section 3.3.1), (2) show how to decompose the task into four subtasks and train four specialized classifiers (Section 3.3.2), and (3) describe our novel features (Section 3.3.3), the manual rule creation process (Section 3.3.4), and our hybrid approach (Section 3.3.5).

3.3.1. The Baseline System—Since the best-performing systems for this task are learning-based [1,3,7,10,13,15,18,23], we will employ a machine learning approach to implement the baseline system.

Creating training instances: Without loss of generality, assume that (e1,e2) is an event-event or event-time pair such that (1) e1 precedes e2 in the associated text and (2) (e1,e2) belongs to one of the three i2b2 temporal relation types. We create one training instance for each event-event/event-time pair in a training document that satisfies the two conditions above, labeling it with the relation type that exists between e1 and e2.

Features: To build a strong baseline, we represent each instance using 92 features modeled after the topperforming temporal relation classification systems developed for TimeBank (e.g., [1]) and the i2b2 corpus (e.g., [3,18,23]), as well as those in the TempEval shared tasks [21,22] (e.g., [7,10,13,15]). Below we divide these features into six categories. The parenthesized numbers represent the number of features belonging to these categories.

Lexical (30): The strings and the head words of e1 and e2; whether e1 and e2 have the same string; word pair formed from the head words of e1 and e2; and word unigrams, bigrams, and trigrams formed from the context within a window of two surrounding e1/e2 [18].

Grammatical (40): The POS tags of the head words of e1 and e2; the POS tags of the five tokens preceding and following e1 and e2; the POS bigram formed from the head word of e1/e2 and its preceding token, the POS tag pair formed from the head words of e1 and e2; the prepositional lexeme of the PP in case e1/e2 is headed by a PP; the prepositional lexeme of the PP in case e1/e2 is governed by a PP; the POS of the head of the VP in case e1/e2 is governed by a VP; whether e1 syntactically dominates e2 [1]; the shortest path from e1 to e2 in the associated syntactic parse tree; pairwise versions of the head word feature and the two prepositional lexeme-based features; the *preposition trace* feature, computed by (1) collecting the list of prepositions along the path from e1/e2 to the root of its syntactic parse trees, and (2) concatenating the resulting lists computed from e1 and e2; the *verb trace* feature, computed in a similar manner, except that we collect the POS tags of the verbs appearing in the corresponding paths. We obtain parse trees and POS tags using the Stanford CoreNLP tool.[3]

Entity attributes (10): The modality, polarity, and event type of e1 and e2 if they are events (if one of them is a time expression, then the class attribute will be set to its class and the rest of them will have the value NULL); pairwise features formed by pairing up the modality values and the type values of e1 and e2; binary features indicating whether e1 and e2 match with respect to type and modality.

[3] http://nlp.stanford.edu/software/corenlp.shtml

Distance (2): The distance between e1 and e2 in the number of tokens; whether they are in the same sentence.

Semantic (7): The subordinating temporal role token of e1/e2 if it appears within a temporal semantic role argument [10]; the first WordNet synset to which ψ/ψ-belongs.

Section creation time related (3): The temporal relation type between ψ/ψ-and the section creation time as predicted by the CRF model described in Section 3.2 (its value can be one of the three types, or NULL if no relation exists); whether ψ-and ψ-have different relation types with SCT.

After creating the training instances, we train a 3-class linear classifier on them using SVM$^{\psi\ \psi\psi\psi\psi\psi\psi\psi\psi\psi}$[19]. We then use it to classify the test instances.

3.3.2. Training Specialized Classifiers—Rather than training just one classifier for classifying all temporal relation instances, Tang et al. [18] show that performance can be improved if we train multiple specialized temporal relation classifiers. For example, we may first divide our training instances based on whether an instance encodes an intra- or inter-sentence temporal relation, and then train two classifiers, one for classifying intra-sentence relations and the other inter-sentence relations.

In fact, Tang et al.'s [18] classifiers are even more specialized than those described in the previous paragraph. They train two intra-sentence classifiers, one for classifying event-event pairs and the other event-time pairs. In addition, they train two inter-sentence classifiers, one for classifying coreferent event pairs and the other event pairs in neighboring sentences.

Given that Tang et al. [18] show initial promise using these four specialized classifiers, we integrate them into our baseline machine learning framework. Below we describe Tang et al.'s method for creating instances for training and testing each of these four specialized classifiers.

Training and applying an intra-sentence event-event classifier: A naïve way to create training/test instances would be to create one training/test instance from each pair of events. This, however, would create a training set with a skewed class distribution, as the negative (i.e., **No-Rel**) instances will significantly outnumber the instances that belong to one of the three relation types shown in Table 1. To address this problem, we create training instances as follows. We create one instance from each event pair in which one of the three relation types exists, labeling the instance with the corresponding relation type. In addition, we create negative instances from two events only if (1) they are adjacent to each other (i.e., there is no intervening event) and (2) no relation exists between them. Test instances are created in the same way as the negative training instances. The implication of the way the test instances are created is that there will be very few temporally related events that are not adjacent to each other in the test set, an assumption that we believe is reasonable though not perfect.

Training and applying an intra-sentence event-time classifier: For the event-time classifier, training and test instances are created in the same way as in the event-event classifier.

Training and applying an inter-sentence classifier for events in adjacent sentences: The difficulty of temporal relation classification tends to increase with the distance between the elements in an event-event or event-time pair. Consequently, Tang et al. [18] consider event-

event pairs only if the two elements involved in a pair are one sentence apart, ignoring event-time pairs entirely since very few of them have a temporal relation.

As mentioned before, one method for creating instances for training and testing would be to create one instance for each event-event/event-time pair. This method, however, suffers from the skewed class distribution of the resulting dataset. Consequently, we employ the following method for creating training and test instances. We create one training instance from every pair of entities that appear in two adjacent sentences and have a temporal relation, assigning it a class value that is the relation type. Negative training instances are created as follows. For every pair of adjacent sentences in a training text, we create four event pairs. The first two pairs are created by pairing the first event from the first sentence with the first event and the last event of the second sentence, respectively. The last two pairs are created by pairing the last event of the first sentence with the first and last events of the second sentence, respectively. We then remove duplicate pairs[4] as well as pairs where the two events have a temporal relation. A negative training instance will then be created from each of the remaining pairs. Test instances are created in a similar manner as the negative training instances. Specifically, for each pair of adjacent sentences in a test text, we first create four event pairs in the same way as we described above, and then create one test instance from each such event pair after removing duplicate pairs. As noted by Tang et al. [18], this way of creating test instances enables us to recover most of the event pairs in a test text that have a temporal relation.

As an example of how test instances are created, consider the following example.

> (4) He has *a left arm graft* placed for access three weeks ago which is used for blood drawing and IV medications. He notes no tenderness, erythema, warmth or exudate from *the passport site*.

In Example 4, since *a left arm graft* is the only event in the first sentence, and *the passport site* is the only event in the second sentence, we create only one test instance from these two sentences, specifically by pairing these two events. Hence, only one test instance will be created from this example.

Training and applying an inter-sentence coreferent event classifier: Unlike the previous classifier, this second inter-sentence classifier places no restriction on how far two events are. However, it handles only a subset of the inter-sentence temporal relations, namely those that are coreferent. The reason for this restriction is that it is intuitively easier to determine the relation type for two coreferent events, since they tend to temporally overlap.

A natural question is: how can we determine whether two events are coreferent? We employ a naïve method: we posit two events as coreferent if and only if they have the same head word.

Next, we describe how the instances for training and testing an inter-sentence coreferent event classifier can be created. We create one training instance from every coreferent event pair in which a temporal relation exists, labeling the instance with the corresponding relation type. We similarly create one negative training instance from every coreferent event pair that does not have any temporal relation. Test instances are created simply by pairing events that are coreferent.

3.3.3. Novel Linguistic Features—In this subsection, we describe our novel features, which will be used to augment the set of basic features to train each of the four specialized

[4]Note that duplicate event pairs arise when one or both sentences have only one event.

classifiers mentioned above. As we will see later in this subsection, some of our features are created based on predicate-argument relations and discourse relations, which are in turn computed using semantic and discourse parsers that have not been trained from clinical text. A natural question is: will these tools provide semantic and discourse annotations that are accurate enough to benefit temporal relation classification for clinical text? Our investigation will shed lights on this question. Nevertheless, our confidence in employing these automatic annotations stems in part from the fact that they will be used in a data-driven manner. For example, one way they will be used is to create additional features for our temporal relation classifiers. Hence, even if not all of these annotations are accurate, the learning algorithm will be able to automatically determine the subset of these annotations that would be useful for temporal relation classification.

Linguistically, our novel features can be divided into four categories:

Quadruple feature: We introduce one quadruple feature by pairing up the tense and class attribute values of e1 with those of e2.

Dependency Relations: A dependency relation is a grammatical relation between two words in a sentence. Dependency relations can be *typed*. For example, a "subject" dependency exists between the verb and its subject in a sentence.

We introduce features computed based on dependency relations obtained via the Stanford CoreNLP tool, motivated by our observation that some dependency relation types are more closely associated with certain temporal relation types than with others. Let us illustrate with an example:

(5) It is aggravated by activity.

In (5), there is an "agent" dependency between the PROBLEM event *aggravated* and the OCCURRENCE event *activity*. In other words, *activity* is the agent of *aggravated*. The reason is that *activity* is the complement of the passive verb *aggravated* introduced by the preposition *by* and performs the action. Intuitively, given a discharge report, if an OCCURRENCE event acts as an agent to a PROBLEM event and there is a temporal relation between them, then it is likely that this temporal relation is **Overlap**.

(6) Psychiatry was *consulted* after the patient was *extubated*.

In (6), there is an adverbial clause modifier dependency between *consulted* and *extubated*, because *extubated* appears in an adverbial clause (headed by *after*) modifying *consulted*. Intuitively, if the two temporally-related events participate in this type of dependency relation and the adverbial clause is headed by *after* and, then it is likely that the temporal relation type is **Before**. In general, given two temporally related events having an adverbial clause modifier dependency, the temporal relation type between them is likely to be **Overlap**, **Before** or **After**, the choice of which can typically be determined by the connective heading the adverbial clause.

Given the potential usefulness of dependency relations for temporal relation classification, we create dependency-based features as follows. For each of the dependency relation types produced by the Stanford parser, we create four binary features: whether ψ/ψ-is the governing entity in the relation, and whether ψ/ψ-is the dependent in the relation.

Predicate-Argument Relations: So far we have exploited lexical and dependency relations for temporal relation classification. Next, we turn to a different type of relations, lexical semantic relations. A lexical semantic relation describes how two words in a sentence are related to each other semantically. For example, a word can be a synonym, antonym,

hypernym, or hyponym of another word. Rather than investigating general lexical semantic relations, we focus on a particular type of lexical semantic relations in this article, predicate-argument relations. A predicate-argument relation is a relation between a predicate and one of its arguments in a sentence. We hypothesize that predicate-argument relations would be useful for temporal relation classification. Consider the following example.

(7) She was *discharged* to *rehab*.

Using SENNA [5], a PropBank-style semantic role labeler, we know that the CLINICAL DEPARTMENT event *rehab* is the A4 argument of the OCCURRENCE event *discharged*. Recall that A4 is the destination point. Hence, we can infer that there is a **Overlap** relation between the OCCURRENCE event and the CLINICAL DEPARTMENT event since the OCCURRENCE event begins at the destination point.

Besides numbered arguments, which have syntactic roles with respect to its predicate, in PropBank-style predicate-argument relations an argument can also have modifier arguments (e.g., CAUSE, PURPOSE, MANNER), which have functional roles with respect to its predicate. Like the numbered arguments, the modifier arguments of a predicate could also inform temporal relation, as shown in the following example.

(8) Discussion should occur with the family about *weaning him* from his medications to make him *more comfortable*.

From SENNA, we know that the predicate *weaning* in the OCCURRENCE event *weaning him* has OCCURRENCE event *more comfortable* in its PURPOSE argument. Intuitively, since an action accomplishes a purpose, we can infer that *weaning him* occurs **Before** *more comfortable.*

So far we have seen that predicate-argument relations are useful for temporal relation classification if a temporal relation exists between a predicate and one of its arguments. The question, then, is: will predicate-argument relations still be useful for temporal relation classification if a temporal relation exists between two different arguments of a predicate? We hypothesize that the answer is affirmative, as illustrated by the following example.

(9) For the past couple of months the patient has had *a non-healing right dorsal foot ulcer* which has been increasing in size and **started** as *a pin hole*.

From SENNA, we know that the predicate **started** has PROBLEM event *a non-healing right dorsal foot ulcer* in its A1 numbered argument and another PROBLEM event *a pin hole* in its Manner modifier argument. Recall that the A1 numbered argument is the one which undergoes the change of state or is being affected by the action. Intuitively, an event that specifies the manner in which a problem started precedes an event that specifies how the problem evolves over time. Hence, we can infer that *a non-healing right dorsal foot ulcer* happens **After** *a pin hole*.

Given the potential usefulness of predicate-argument relations involving one or two arguments of a predicate, we create binary features based on predicate-argument relations as follows. We create one binary feature from each predicate-argument relation, setting its value to 1 if the predicate and the type of the argument encoded by this feature are present in the instance under consideration.[5] In addition, we create one binary feature from each pair of arguments of a predicate, setting its value to 1 if the predicate and the types of the two arguments encoded by this feature are present in the instance under consideration. Note, however, that there is a restriction in the creation of these features: any feature that involves

[5]Note that SENNA was trained on PropBank, where an argument type does not have a subtype.

a modifier argument that has type other than DIRECTIONAL, MANNER, TEMPORAL, and CAUSE, will be discarded. The reason is based on our observation that predicate-argument relations involving modifier arguments that do not belong to any of these four types were identified with a lot of noise, presumably because SENNA was trained on Newswire articles, not clinical notes.

Discourse Relations: A discourse relation (also known as a rhetorical relation) specifies how two segments of a discourse are logically connected to each other. For example, the current sentence has an elaboration relation with the previous one because it elaborates what a discourse relation is. Discourse relations such as causation, elaboration and enablement could aid in tracking the temporal progression of the discourse [8]. Hence, unlike syntactic dependencies and predicate-argument relations through which we can identify *intra-sentential* temporal relations, discourse relations can potentially be exploited to discover both *inter-sentential* and *intra-sentential* temporal relations. However, no recent work has attempted to use discourse relations for temporal relation classification. In this subsection, we examine whether we can improve a temporal relation identifier via *explicit* and *implicit* PDTB-style discourse relations automatically extracted by Lin et al.'s [9] end-to-end discourse parser.

Let us first review PDTB-style discourse relations. Each relation is represented by a triple (*Arg1, sense, Arg2*), where *Arg1* and *Arg2* are its two arguments and *sense* is its sense/type. A discourse relation can be explicit or implicit. An explicit relation is triggered by a discourse connective. On the other hand, an implicit relation is not triggered by a discourse connective, and may exist only between two consecutive sentences. Generally, implicit relations are much harder to identify than their explicit counterparts.

Next, to motivate why discourse relations can be useful for temporal relation classification, we use three examples (see Table 4), two involving an explicit relation (Examples (10) and (11)) and one involving implicit relation (Example (12)). For convenience, both sentences are also annotated using Lin et al.'s [9] end-to-end PDTB-style discourse parser, which marks up the two arguments with the Arg1 and Arg2 tags and outputs the relation sense next to the beginning of Arg2.

In (10), we aim to determine the relation type between the TREATMENT event *operation* and the OCCURRENCE event *benign convalescence*. The parser determines that an ASYNCHRONOUS explicit relation triggered by the discourse connective *thereafter* exists between the two sentences, suggesting that the two events are likely to have an asynchronous temporal relation type such as **Before** or **After**. By considering the discourse connective *thereafter*, we can infer that the correct temporal relation type is **Before**.

In (11), we aim to determine the temporal relation type between two TREATMENT events, *coronary atherectomy* and *directional atherectomy*. The parser determines that a RESTATEMENT explicit relation exists between the two sentences. Intuitively, two temporally linked TREATMENT events within different discourse units connected by the RESTATEMENT relation imply some sort of synchronicity in their temporal relation, meaning that the relation type is **Overlap**.

Example (12) has two temporally related TREATMENT events contained within separate discourse arguments, *her additional therapy* and *further available treatment*, where the second argument is annotated as an implicit RESTATEMENT of the first by the parser. As in (11), we infer that the temporal relation type is **Overlap** by relying on the greater chance of temporal synchronicity between events mentioned in sentences where one is a restatement of the other.

Given the potential usefulness of discourse relations for temporal relation classification, we create four features based on discourse relations. In the first feature, if e1 is in Arg1, e2 is in Arg2, and Arg1 and Arg2 possess an explicit relation with sense s, then its feature value is s; otherwise its value is NULL. In the second feature, if e2 is in Arg1, e1 is in Arg2, and Arg1 and Arg2 possess an explicit relation with sense s, then its feature value is s; otherwise its value is NULL. The third and fourth features are computed in the same way as the first two features, except that they are computed over implicit rather than explicit relations.

3.3.4. Manual Rule Creation—As noted before, we adopt a hybrid learning-based and rule-based approach to temporal relation classification. Hence, in addition to training a temporal relation classifier, we manually design a set of rules in which each rule returns a temporal relation type for a given test instance. We hypothesize that a rule-based approach can complement a purely learning-based approach, since a human could combine the available features into rules using commonsense knowledge that may not be accessible to a learning algorithm.

The design of the rules is partly based on intuition and partly data-driven: we first use our intuition to come up with a rule and then manually refine it based on the observations we made on the i2b2 training documents. Note that the test documents are reserved for evaluating final system performance. We order these rules in decreasing order of accuracy, where the accuracy of a rule is defined as the number of times the rule yields the correct temporal relation type divided by the number of times it is applied, as measured on the training documents. A new instance is classified using the first applicable rule in the ruleset. In total there are hand-crafted 665 rules. Some of them were shown in the previous subsection when we motivated each feature type with examples. To enable the reader to gain a better understanding of these rules, we listed 20 of them in the appendix. Our final ruleset can be accessed via a web link.[6]

3.3.5. Combining Rules and Machine Learning—We investigate two ways to combine the hand-crafted rules and the machine-learned classifier.

In the first method, we employ all of the rules as additional features for training the classifier. The value of each such feature is the temporal relation type predicted by the corresponding rule. The second method operates as follows. Gven a test instance, we first apply to it the ruleset composed only of rules that are at least 80% accurate. If none of the rules is applicable, we classify it using the classifier employed in the first method.

4. Results and Discussion

Table 5 shows our results for the TLINK track that are obtained from linking event/timex3 entity pairs and events with section creation times. Each row corresponds to a different system. More specifically, these systems employ the same method for classifying the TLINK between an event/time expression with the section creation time (i.e., the method described Section 3.2), differing only in terms of the method used for classifying the TLINK between two entities.

Row 1 of Table 5 shows the results of employing the learning-based baseline system described in Section 3.3.1 for classifying the TLINK between two entities. Recall that this baseline system trains a single temporal relation classifier using 92 features that are modeled after the top-performing temporal relation classification systems developed for TimeBank. One thing that we left unspecified when describing the baseline system in Section 3.3.1 is

[6] https://docs.google.com/file/d/0B5q--YIRwCmBY2RyU3E1clJOOFE/edit?pli=1

how the test instances should be created. To ensure a fair comparison among all these systems, we evaluate them on the same set of test instances. Specifically, we evaluate them on the test instances created using the method described in Section 3.3.2. Note that the test instances created by this method represent a subset of all the entity pairs in the test documents. Hence, an implicit assumption underlying this test instance creation method is that any entity pair for which no test instance is generated is classified as having no temporal relation. As we can see, our baseline system achieves an F-measure of 65.2%.

Row 2 of Table 5 shows the results when the single classifier in row 1 is replaced with the four specialized classifiers described in Section 3.3.2. In comparison to the baseline, F-measure increases by 1.9 percentage points to 67.1%, as a result of large increases in recall accompanied by smaller drops in precision. These results demonstrate the effectiveness of using specialized classifiers for different types of event pairs.

Row 3 of Table 5 shows the results when the feature set employed by the system in row 2 is augmented with the four types of novel features that we proposed in Section 3.3.3. In comparison to the system in row 2, we see that F-measure increases by 1.0 percentage point to 68.1%, owing to a 1.6% improvement in precision and a 0.1% drop in recall.

Row 4 of Table 5 shows the results of the system where we use all the hand-crafted rules as additional features for training the system in row 3. This architecture corresponds to the first method for combining rules and machine learning in Section 3.3.5. In comparison to row 3, we see that when these rules are used as additional features, F-measure increases by 0.5 percentage points to 68.6, owing to a 1.2% improvement in precision accompanied by a 1.0% drop in recall.

Row 5 of Table 5 shows the results of the system created by combining hand-crafted rules and machine learning using the second method described in Section 3.3.5. Recall that in this system, a test instance is first classified by the high-accuracy rules (i.e., the 426 rules whose accuracy is at least 80%), and if none of the rules is applicable, it will be classified using the system in row 4. In comparison to row 4, we see that F-measure increases by 0.7 percentage points to 69.3%, as a result of a nearly 4% improvement in precision accompanied by a nearly 4% drop in recall. It should perhaps not be surprising to see that precision increases, since we attempt to classify a test instance using high-accuracy rules prior to applying the machine-learned specialized classifiers. The F-measure score achieved by this system is the best result reported to date on this dataset.

Since our best-performing system is a pipeline architecture composed of a ruleset and a machine-leaned classifier, a natural question is: how much does each component contribute to overall performance? To answer this question, we remove the machine-learned component from the best-performing system, using only the high-accuracy rules for classifying the test instances. Results of this experiment are shown in row 6 of Table 5. As we can see, these high-accuracy rules are indeed highly accurate, achieving a precision of nearly 89% on the test set. However, because of the fairly low recall (41.7), F-measure drops by 12.6 percentage points in comparison to the best-performing system. This performance difference is also what is contributed by the machine-learned classifier to performance of the best-performing system.

5. Conclusions

We investigated a knowledge-rich, hybrid approach to the TLINK extraction track in the 2012 i2b2 challenge. Experiments on the i2b2 corpus demonstrated that our approach achieves an F-score of 69.3, which is the best result reported to date on this dataset. To

stimulate research on this task, we will make our complete set of handcrafted rules publicly available.

Since our approach currently uses the predicate-argument relations and the discourse relations generated by semantic and discourse parsers that are not trained on clinical text, we believe that a promising way to improve our approach would be to replace these parsers with those trained on clinical text.

Appendix

Below we show 20 of our hand-crafted rules. To understand how to interpret these rules, let us take Rule 1 as an example. Rule 1 says that if TREATMENT event1 and PROBLEM event2 appear in two discourse segments of the same sentence that are the two arguments of a CONDITION explicit discourse relation, then an **After** temporal relation exists between the two events. The remaining rules can be interpreted in a similar manner.

Rule1: if sameSentence=TRUE &&
entity1.class=TREATMENT &&
entity2.class=PROBLEM &&
discourseExplicitRelationArg1ConditionArg2(entity1, entity2)
then infer relation=**After**;

Rule2: if consecutiveSentence=TRUE &&
entity1.class=PROBLEM && entity1.modality=FACTUAL &&
entity2.class=PROBLEM && entity2.modality=FACTUAL &&
discourseImplicitRelationArg1RestatementArg2(entity1, entity2)
then infer relation=**Simultaneous**;

Rule3: if consecutiveSentence=TRUE &&
entity1.class=EVIDENTIAL && entity1.modality=FACTUAL &&
entity2.class=PROBLEM && entity2.modality=FACTUAL &&
entity1.text.contains("report") &&
(entity2.precededBy(",") || entity2.precededBy("or")) &&
discourseExplicitRelationArg1EntRelArg2(entity1, entity2)
then infer relation=**Overlap_After**;

Rule4: if consecutiveSentence=TRUE &&
entity1.class=TREATMENT && entity1.modality=FACTUAL &&
entity2.class=TREATMENT && entity2.modality=FACTUAL &&
discourseExplicitRelationArg 1CauseArg2(entity1, entity2)
then infer relation=**Simultaneous**;

Rule5: if consecutiveSentence=TRUE &&
entity1.class=EVIDENTIAL && entity1.modality=FACTUAL &&
entity2.class=EVIDENTIAL && entity2.modality=FACTUAL &&

discourseExplicitRelationArg1ConjunctionArg2(entity1, entity2)

then infer relation=**Simultaneous**;

Rule6: if consecutiveSentence=TRUE &&

entity1.class=TREATMENT && entity1.modality=FACTUAL &&

entity2.class=TREATMENT && entity2.modality=FACTUAL &&

discourseImplicitRelationArg1ConjunctionArg2(entity1, entity2)

then infer relation=**Simultaneous**;

Rule7: if consecutiveSentence=TRUE &&

entity1.class=OCCURRENCE && entity1.modality=FACTUAL &&

entity2.class=TREATMENT && entity2.modality=FACTUAL &&

entity1.text.equals("admission") &&

entity2.governVerbWord.equals("continued)

discourseImplicitRelationArg1EntRelArg2(entity1, entity2)

then infer relation=**Overlap_After**;

Rule8: if sameSentence=TRUE &&

entity1.class=CLINICAL_DEPT && entity1.modality=FACTUAL &&

entity2.class=TREATMENT && entity2.modality=FACTUAL &&

entity1.precededBy("in") &&

discourseExplicitRelationArg1ConjunctionArg2(entity1, entity2)

then infer relation=**During_Inv**;

Rule9: if entity1.class=PROBLEM &&

entity2.class=OCCURRENCE &&

discourseExplicitRelationArg1SynchronyArg2(entity1, entity2)

then infer relation=**Simultaneous**;

Rule10: if consecutiveSentence =TRUE &&

entity1.class=PROBLEM &&

entity2.class=PROBLEM &&

discourseExplicitRelationArg1CauseArg2(entity1, entity2)

then infer relation=**Overlap**;

Rule11: if sameSentence =TRUE &&

entity1.class=OCCURRENCE &&

entity2.class=TREATMENT &&

entity1.text.equals("reduce") &&

entity1.hasArgument1=TRUE &&

entity1.srlArgument1.contains(entity2)

then infer relation=**Ends**;

Rule12: if sameSentence =TRUE &&
entity1.class=OCCURRENCE &&
entity2.class=DATE &&
entity1.hasArgument3=TRUE &&
entity1.srlArgument3.contains(entity2) &&
entity2.precededBy("from") &&
then infer relation=**Begun_By**;

Rule13: if entity1.class=DURATION &&
entity2.class=OCCURRENCE &&
entity2.hasTemporalArgument=TRUE &&
entity2.srlTemporalArgument.contains(entity1)
then infer relation=**During_Inv**;

Rule14: if entity1.class=TREATMENT &&
entity2.class=CLINICAL_DEPT &&
entity1.isInArgument2=TRUE && entity2.isInArgument2=TRUE &&
dependency_prep_in(entity1, entity2)
then infer relation=**During**;

Rule15: if entity1.class=PROBLEM &&
entity2.class=DURATION &&
entity1.hasTemporalArgument=TRUE &&
entity1.srlTemporalArgument.contains(entity2)
then infer relation=**During**;

Rule16: if entity1.class=EVIDENTIAL && entity1.modality=FACTUAL &&
entity2.class=TEST && entity2.modality=FACTUAL &&
entity1.text.equals("showed") &&
entity1.hasArgument1=TRUE &&
entity1.srlArgument1.contains(entity2) &&
isConsecutive(entity1, entity2)
then infer relation=**Overlap_After**;

Rule17: if entity1.class=TEST && entity1.modality=FACTUAL &&
entity2.class=PROBLEM && entity2.modality=FACTUAL &&
srlVerb.text.equals("demonstrated") &&
entity1.isInArgument0=TRUE &&
entity2 isInArgument1=TRUE

then infer relation=**Overlap_After**;

Rule18: if entity1.class=EVIDENTIAL && entity1.modality=FACTUAL &&
entity2.class=PROBLEM && entity2.modality=FACTUAL &&
entity1.text.endsWith("ed") &&
dependency_dobj(entity1, entity2) &&
entity1.hasArgument1=TRUE &&
entity1.srlArgument1.contains(entity2) &&
then infer relation=**Overlap_After**;

Rule19: if entity1.class=TREATMENT && entity1.modality=FACTUAL &&
entity2.class=PROBLEM && entity2.modality=FACTUAL &&
srlVerb.text.equals("causing") &&
entity1.isInArgument0=TRUE &&
entity2.isInArgument1=TRUE &&
then infer relation=**Before_Overlap**;

Rule20: if entity1.class=OCCURRENCE &&
entity2.class=CLINICAL_DEPT &&
dependency_prep_to(entity1, entity2)
then infer relation=**Begins**;

References

1. Chambers, N.; Wang, S.; Jurafsky, D. Proceedings of ACL Companion Volume: Proceedings of the Demo and Poster Sessions. 2007. Classifying temporal relations between events; p. 173-176.
2. Chambers, N.; Jurafsky, D. Proceedings of EMNLP. 2008. Jointly combining implicit constraints improves temporal ordering; p. 698-706.
3. Cherry C, Zhu X, Martin J, de Bruijn B. À la Recherche du Temps Perdu: extracting temporal relations from medical text in the 2012 i2b2 NLP challenge. Journal of the American Medical Informatics Association. 2013
4. Chklovski, T.; Pantel, P. Proceedings of EMNLP. 2004. Verbocean: Mining the web for fine-grained semantic verb relations; p. 33-40.
5. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa PP. Natural language processing (almost) from scratch. Journal of Machine Learning Research. 2011; 12:2493–2537.
6. Finkel J, Dingare S, Manning C, Nissim M, Alex B, Grover C. Exploring the boundaries: gene and protein identification in biomedical text. BMC Bioinformatics. 2005; 6:S5. [PubMed: 15960839]
7. Ha, EY.; Baikadi, A.; Licata, C.; Lester, J. Proceedings of the 5th International Workshop on Semantic Evaluation. 2010. NCSU: Modeling temporal relations with markov logic and lexical ontology; p. 341-344.
8. Hitzeman, J.; Moens, M.; Grover, C. Proceedings of the 7th Conference of the EACL. 1995. Algorithms for analysing the temporal structure of discourse; p. 253-260.
9. Lin Z, Ng HT, Kan MY. A PDTB-styled end-to-end discourse parser. Natural Language Engineering (to appear). 2013
10. Llorens, H.; Saquete, E.; Navarro, B. Proceedings of the 5th International Workshop on Semantic Evaluation. 2010. TIPSem (English and Spanish): Evaluating CRFs and semantic roles in TempEval-2; p. 284-291.

11. Mani, I.; Verhagen, M.; Wellner, B.; Lee, CM.; Pustejovsky, J. Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. 2006. Machine learning of temporal relations; p. 753-760.
12. McDonald R, Pereira F. Identifying gene and protein mentions in text using conditional random fields. BMC Bioinformatics. 2005; 6:S6. [PubMed: 15960840]
13. Min, C.; Srikanth, M.; Fowler, A. Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). 2007. LCC-TE: A hybrid approach to temporal relation identification in news text; p. 219-222.
14. Prasad, R.; Dinesh, N.; Lee, A.; Miltsakaki, E.; Robaldo, L.; Joshi, A.; Webber, B. Proceedings of the 6th International Conference on Language Resources and Evaluation. 2008. The penn discourse tree-bank 2.0.
15. Puşcaşu, G. Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). 2007. WVALI: Temporal relation identification by syntactico-semantic analysis; p. 484-487.
16. Pustejovsky, J.; Hanks, P.; Sauri, R.; See, A.; Day, D.; Ferro, L.; Gaizauskas, R.; Lazo, M.; Setzer, A.; Sundheim, B. Corpus Linguistics. 2003. The TimeBank corpus; p. 647-656.
17. Sun W, Rumshisky A, Uzuner O. Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. Journal of American Medical Informatics Assoication. 201310.1136/amiajnl-2013-001628
18. Tang, B.; Wu, Y.; Jiang, M.; Chen, Y.; Denny, J.; Xu, H. Proceedings of the 6th i2b2 Shared Tasks and Workshop on Challenges in Natural Langauge Processing for Clincial Data. 2012. Extracting temporal information from clinical text — Vanderbilt's system for the 2012 i2b2 NLP challenge.
19. Tsochantaridis, I.; Hofmann, T.; Joachims, T.; Altun, Y. Proceedings of the 21st ICML. 2004. Support vector machine learning for interdependent and structured output spaces; p. 104-112.
20. UzZaman, N.; Allen, JF. Proceedings of the Fifth International Workshop on Semantic Evaluations (SemEval-2010). 2010. TRIPS and TRIOS system for TempEval-2: Extracting temporal information from text; p. 276-283.
21. Verhagen, M.; Gaizauskas, R.; Schilder, F.; Hepple, M.; Katz, G.; Pustejovsky, J. Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007). 2007. SemEval-2007 Task 15: TempEval temporal relation identification; p. 75-80.
22. Verhagen, M.; Sauri, R.; Caselli, T.; Pustejovsky, J. Proceedings of the 5th International Workshop on Semantic Evaluation. 2010. SemEval-2010 Task 13: TempEval-2; p. 57-62.
23. Xu, Y.; Wang, Y.; Liu, T.; Tsujii, J.; Chang, E. Proceedings of the 6th i2b2 Shared Tasks and Workshop on Challenges in Natural Langauge Processing for Clincial Data. 2012. An end-to-end system to identify temporal relation in discharge summaries.
24. Zhou G, Shen D, Zhang J, Su J, Tan S. Recognition of protein/gene names from text using an ensemble of classifiers. BMC Bioinformatics. 2005; 6:S7. [PubMed: 15960841]

Highlights

- Detecting and classifying TLINKs using:
  - PropBank-style predicate-argument relations, and
  - Discourse relations.
- A hybrid system architecture employing both rules and machine-learning.
- F-score of 69.3 on TLINK detection and classification.

Table 1

The three possible types of temporal relations defined on an event-section creation time, an event-event pair or an event-time pair in the i2b2 corpus. Each relation is defined on an ordered pair (e1,e2), where e1 and e2 can each be a section creation time, an event or a time expression. The "Total" and "%" column shows the number and percentage of instances annotated with the corresponding relation type in the corpus, respectively, and the "E-E" and "E-T" columns show the breakdown by the number of event-event pairs and event-time pairs.

Relation Type	Description	Event-SCT Total (%)	Entity-Entity Total (%)	Entity-Entity	
				E-E	E-T
Overlap	e_1 and e_2 happen at the same time but not exactly	1349 (8.4)	11389 (29.9)	9102	2287
Before	e_1 happens before e_2 in time	14072 (87.7)	3304 (8.6)	2806	498
After	e_1 happens after e_2 in time	630 (3.9)	2746 (7.2)	2348	398
No-Rel	e_1 and e_2 have no relation	–	20629 (54.2)	16160	4469

Table 2

Description of the general features for EVENT/TIMEX3-SCT TLINK type prediction. w_i denotes the current word

Type	Symbolic Notation
Word Features	w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}
Lexemes	lex_i, lex_{i-1}, lex_{i-2}, lex_{i+1}, lex_{i+2}
Lexeme Bigrams	$lex_i + lex_{i-1}$, $lex_{i-1} + lex_{i-2}$, $lex_i + lex_{i+1}$, $lex_{i+1} + lex_{i+2}$
POS	POS_i, POS_{i-1}, POS_{i-2}, POS_{i+1}, POS_{i+2}
POS Bigrams	$POS_i + POS_{i-1}$, $POS_{i-1} + POS_{i-2}$, $POS_i + POS_{i+1}$, $POS_{i+1} + POS_{i+2}$
Affixes	Prefixes of up to a length of 3, suffixes of up to a length of 3
Word Shape	wSI_i, $wSII_i$, $wSII_{i-1}$, $wSII_{i-2}$, $wSII_{i+1}$, $wSII_{i+2}$
Word Shape II Bigrams	$wSII_i + wSII_{i-1}$, $wSII_{i-1} + wSII_{i-2}$, $wSII_i + wSII_{i+1}$, $wSII_{i+1} + wSII_{i+2}$
Other	$isUpperCase_i$, $beginsWithUpperCase_i$, $beginsWithUpperCaseFollowedByLowerCase_i$, $isCaseMixture_i$, $hasDigit_i$, $isSingleDigit_i$, $isDoubleDigit_i$, $isInteger_i$, $isRealNumber_i$, $hasHyphen_i$, $isAlphaNumeric_i$

Table 3

Description of the phrase-based feature used for EVENT/TIMEX3-SCT TLINK type prediction

Phrase Type	Symbolic Notation
Head Verb Phrase	verb-word$_h$, verb-POS$_h$
Governing Verb Phrase	verb-word$_g$, verb-POS$_g$
Governing Prepositional Phrase	prep-word$_g$
Token-based Phrasal Attributes	If EVENT {type, polarity, modality} else if TIMEX3 {type, mod}

Table 4

Examples illustrating the usefulness of discourse relations for temporal relation classification.

(10) {Arg1 At *operation*, there was no gross adenopathy, and it was felt that the tumor was completely excised. Arg1} {Arg2 The patient {Conn ASYNCHRONOUS thereafter Conn} had a *benign convalescence*. Arg2}
(11) {Arg1 Coronary angiography demonstrated ongoing benefit of the initial *coronary atherectomy*. Arg1} {Conn RESTATEMENT Specifically Conn}, {Arg2 there was no decrease in the initial gain that she achieved after *directional atherectomy*. Arg2}
(12) {Arg1 Given this we were unable to offer *her additional therapy* Arg1}. {Arg2 RESTATEMENT Hematology/Oncology service at Mediplex Rehab Hospital confirmed her grim prognosis and expective survival in terms of weeks to months with no *further available treatment* Arg2}..

Table 5

Precision, Recall and F measure of different systems on the test set.

		Precision	Recall	F measure
1	Baseline System (Single classifier, baseline features)	62.1	68.6	65.2
2	+Four Specialized Classifiers	58.3	79.1	67.1
3	+Linguistic Features	59.9	79.0	68.1
4	+Rules as Features	61.1	78.0	68.6
5	+Rules	65.0	74.3	69.3
6	Rules with accuracy at least 80% only	88.8	41.7	56.7