Temporal Relation Identification and Classification in Clinical Notes

Jennifer D'Souza and Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{jld082000,vince}@hlt.utdallas.edu

ABSTRACT

We examine the task of temporal relation classification for the clinical domain. Our approach to this task departs from existing ones in that it is (1) *knowledge-rich*, employing sophisticated knowledge derived from semantic and discourse relations, and (2) *hybrid*, combining the strengths of rule-based and learning-based approaches. Evaluation results on the i2b2 Clinical Temporal Relations Challenge corpus show that our approach yields a 15–21% and 6–13% relative reduction in error over a state-of-the-art learning-based baseline system when gold-standard and automatically identified temporal relations are used, respectively.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text Analysis

General Terms

Algorithms, Experimentation

Keywords

temporal relations, temporal reasoning, bioinformatics

1. INTRODUCTION

Temporal relation classification, one of the most important temporal information extraction (IE) tasks, involves classifying a given event-event pair or event-time pair in a text as one of a set of predefined temporal relations. The creation of the TimeBank corpus [14], as well as the organization of the TempEval-1 [19] and TempEval-2 [20] evaluation exercises, have facilitated the evaluation of temporal relation classification systems for the news domain.

Our goal in this paper is to advance the state of the art in temporal relation classification. While virtually all previous work on this task has focused on the news domain, we work with a relatively unexplored domain, the *clinical domain*, using the i2b2 Clinical Temporal Relations Challenge corpus (henceforth the *i2b2 corpus*).[1] To date, this corpus is only accessible to and has only been experimented on by the participants of the Challenge (henceforth the *shared task*).

Our work differs from existing work with respect to both the *complexity* of the task we are addressing and the *approach* we adopt. Regarding task complexity, rather than focus on *three* temporal relations as in the shared task (see Section 2 for more information), we address an arguably more challenging version of the task where we consider all the 12 relations originally defined in the i2b2 corpus.

Our approach to temporal relation classification can be distinguished from existing 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. Existing approaches have relied primarily on morpho-syntactic features, as well as a few semantic features extracted from WordNet synsets and VerbOcean's [3] semantic relations. On the other hand, we propose not only novel lexical and grammatical features, but also sophisticated features involving semantics and discourse. Most notably, we propose (1) discourse features encoding Penn Discourse TreeBank (PDTB) style [12] discourse relations and (2) semantic features encoding a variety of semantic relations. The latter include PropBank-style predicate-argument relations as well as relations extracted from the Merriam-Webster dictionary.

Second, while the vast majority of existing approaches to temporal relation classification 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 our hypothesis that better decision rules can 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 were shown to underperform learning-based approaches [10], to our knowledge they have not been used in combination with learning-based approaches. Moreover, while the rules employed in previous work were created based on intuition (e.g., Mani et al. [10], Puşcaşu [13]), our rules are created in a *data-driven* manner via a manual inspection of the annotated temporal relations in the i2b2 corpus.

We evaluate our knowledge-rich, hybrid approach to temporal relation classification in two settings. In the first setting, we assume that we are given event-event and event-time pairs that are known to belong to one of the 12 predefined temporal relations in the i2b2 corpus, and hence the

[1]See `https://www.i2b2.org/` for more information.

BCB '13, September 22 – 25, 2013, Washington, DC, USA

task is to label each pair with one of these 12 relation types. To make things more challenging, however, we assume in the second setting that we are given event-event and event-time pairs that *may* or *may not* belong to one of the 12 relation types. Hence, the task in this setting involves both *identifying* and *classifying* temporal relations. For this task, we first employ a *relation identification* system to determine whether a pair has a relation, and then use the same *relation classification* system as the one in the first setting to classify all and only those pairs that are determined to have a relation by the identification system. Conducting experiments with both settings can enable us to determine how much performance deterioration can be attributed to *identifying* rather than *classifying* temporal relations.

Experiments on the i2b2 corpus show the effectiveness of our approach: under the first and the second settings, it yields a 15–21% and 6–13% relative error reduction respectively over a state-of-the-art learning-based baseline system.

To our knowledge, we are the first to (1) report results for the 12-class temporal relation classification task on the i2b2 corpus; (2) successfully employ automatically computed predicate-argument relations and PDTB-style discourse relations to improve performance on this task; and (3) show that a hybrid approach to this task can yield better results than either a rule-based or learning-based approach. In addition, we release the complete set of rules that we mined from the i2b2 corpus and used in our rule-based approach[2], hoping that our insights into how features can be combined as decision rules can benefit researchers interested in this task.

2. CORPUS

For evaluation, we use 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). Each summary is composed of two sections. The first section was created when the patient was admitted and reports History of Present Illness (i.e., her clinical history). The second section was created when the patient was discharged from the hospital and reports Hospital Course. In each summary, the *events*, *times*, and their *temporal relations* are marked up. An event, which can be a verb phrase, an adjective phrase, a noun phrase, or sometimes an adverb that semantically refers to clinically relevant patient-related actions, contains various attributes, including the *type* of event[3], *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. A temporal relation can be an *anchor* relation, which anchors an event to a time expression (as in sentence (1)), or an *order* relation, which orders two events (as in sentence (2)).

(1) He was ready for *discharge* home on *postoperative day three*.
(2) She has not *complained* of any *fever*.

Each temporal relation has a *type*. For example, the relation defined on *discharge* and *postoperative day three* in (1) has type **Simultaneous**, whereas the relation defined on *complained* and *fever* in (2) has type **Overlap_After**. A temporal relation is defined on an *ordered* pair: in (2), the pair (*complained*, *fever*) has type **Overlap_After**, whereas the pair (*fever*, *complained*) has type **Before_Overlap**.

12 relation types are defined and used to annotate the temporal relations in the i2b2 corpus. Table 1 provides a brief description of these relation types and the relevant statistics.

As mentioned in the introduction, our approach will be evaluated in two settings: in the first setting, we employ gold-standard temporal relations, and in the second one, we employ automatically identified temporal relations. In both settings, we follow the i2b2 Temporal Challenge Tlink track and assume that gold events and time expressions are given.

Unlike the shared task, which focuses on three broad relation types (**Overlap'**, **Before'**, **After'**), we report results on 12 relation types. Note that the three broad relation types are created by merging "similar" relation types as follows: (1) **Overlap'** is composed of **Overlap**, **Simultaneous**, and **During**; (2) **Before'** is composed of **Before**, **Before_Overlap**, and **Ended_By**; and (3) **After'** is composed of **After** and **Begun_By**. Each instance from the remaining four relations is merged into one of the three broad relation types by inverting the order of its elements. For example, if a relation instance (e_1, e_2) is annotated as **Ends**, it is first replaced with the instance (e_2, e_1) with class **Ended_By** and then re-labeled as **Before'**. Thus, classifiers developed for the shared task are only presented with test instances belonging to one of the three broad relation types. On the other hand, our 12-class task is arguably more challenging, since our system has to distinguish not only a relation type from its inverse, but also between "similar" relation types.

3. BASELINE TEMPORAL RELATION CLASSIFIER

Since the best-performing systems for temporal relation classification for both the news and clinical domains are learning-based, we will employ a learning-based system as our baseline. Below we describe how we train this baseline.

Without loss of generality, assume that (e_1,e_2) is an event-event/event-time pair such that (1) e_1 precedes e_2 in the associated text and (2) (e_1,e_2) belongs to one of the 12 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 e_1 and e_2.

To build a strong baseline, we represent each instance using 67 features modeled after the top-performing temporal relation classification systems on TimeBank (e.g., Chambers et al. [2], Mani et al. [10]) and the i2b2 corpus (e.g., Tang et al. [17], Xu et al. [21]), as well as those in the TempEval shared tasks (e.g., Ha et al. [6], Llorens et al. [9], Min et al. [11], Puşcaşu [13]). These features can be divided into six categories, as described below.

Lexical (17). Word unigrams, bigrams, and trigrams formed from the context within a window of two words surrounding e_1/e_2, the strings and the head words of e_1 and e_2, and whether e_1 and e_2 have the same string.

Grammatical (33). The POS tags of the head words of e_1 and e_2, the POS tags of the five tokens preceding and following e_1 and e_2, the POS bigram formed from the head

[2]Downloadable from `http://www.hlt.utdallas.edu/~jld082000/temporal-relations/`

[3]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*).

Id	Relation	Description	Total (%)	E-E	E-T
1	**Simultaneous**	e_1 and e_2 happen at the same time or are temporally indistinguishable	4589 (32.5)	3551	1038
2	**Overlap**	e_1 and e_2 temporally overlap but do not happen at the same time	5681 (40.2)	4713	968
3	**Before**	e_1 happens before e_2 in time	1572 (11.1)	1439	133
4	**After**	e_1 happens after e_2 in time	577 (4.1)	497	80
5	**Before_Overlap**	e_1 happens prior to and continues at the time of e_2	506 (3.6)	473	33
6	**Overlap_After**	e_1 overlaps with and happens after e_2 begins	1642 (11.6)	1584	58
7	**During**	e_1 persists throughout duration e_2	386 (2.7)	220	166
8	**During_Inv**	e_2 persists throughout duration e_1	640 (4.5)	522	118
9	**Begins**	e_1 marks the beginning of e_2	782 (5.5)	591	191
10	**Begun_By**	e_2 marks the beginning of e_1	204 (1.4)	65	139
11	**Ends**	e_1 marks the end of e_2	318 (2.3)	197	121
12	**Ended_By**	e_2 marks the end of e_1	434 (3.1)	293	141

Table 1: The 12 temporal relation types in the i2b2 corpus. Each relation type is defined on an ordered pair (e_1,e_2), where e_1 and e_2 can each be an event or a time. The "Total" and "%" columns show 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.

word of e_1/e_2 and its preceding token, the POS tag pair formed from the head words of e_1 and e_2, the prepositional lexeme of the prepositional phrase (PP) if e_1/e_2 is headed by a PP, the prepositional lexeme of the PP if e_1/e_2 is governed by a PP, the POS of the head of the verb phrase (VP) if e_1/e_2 is governed by a VP, whether e_1 syntactically dominates e_2 (Chambers et al. 2007), and the shortest path from e_1 to e_2 in the associated syntactic parse tree. We obtain parse trees and POS tags using the Stanford CoreNLP tool.[4]

Entity attributes (8). The type, modality, and polarity of e_1 and e_2 if they are events (if one of them is a time expression, then its modality and polarity attributes will have the value NULL), pairwise features formed by pairing up the type and modality attribute values of e_1/e_2.

Distance (2). The distance between e_1 and e_2 in number of tokens, whether e_1 and e_2 in the same sentence.

Semantic (4). The subordinating temporal role token of e_1/e_2 if it appears within a temporal semantic role argument (Llorens et al. 2010), and the first WordNet synset to which e_1/e_2 belongs.

Section creation time (SCT) related (3). The temporal relation type between e_1/e_2 and the creation time of the section in which it appears [its value can be one of the 3 relation types (i.e., **Before**, **After**, or **Overlap**) or NULL if no relation exists], and whether e_1 and e_2 have different relation types with the SCT.

3.1 Training Specialized Classifiers

After creating the training instances, we can train a temporal relation classifier on them using an off-the-shelf learner and use the resulting classifier to classify the test instances. However, Tang et al. [17], the best performer in the shared task, shows that performance can be improved by training four specialized classifiers rather than just one for classifying all temporal relation instances. Specifically, they train two intra-sentence classifiers, one for classifying event-event pairs and the other event-time pairs. They also train two inter-sentence classifiers, one for classifying coreferent event pairs and the other for classifying event pairs in neighboring sentences.

Since Tang et al.'s [17] approach looks promising, we integrate their four specialized classifiers into our machine learning framework in order to build a strong baseline. Below we describe Tang et al.'s method for creating instances for training and testing each of the four specialized classifiers.

Training and applying an intra-sentence event-event classifier. A naive 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-Relation) instances will significantly outnumber the instances that belong to one of the 12 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 12 relation types exists, labeling the instance with one of the relation types. 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. During testing, test instances are created in the same way as the negative training instances.

Training and applying an intra-sentence 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. [17] 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, skews the 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 event-event/event-time pair whose elements (1) have a temporal relation and (2) occur in adjacent sentences, and assign it a class value that is the relation type. In addition, we create one negative training instance from each pair of main events that appear in adjacent sentences, where the main events of a sentence are simply the first and last events

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

of a sentence. Test instances are created in the same way as the negative training instances.

Training and applying an inter-sentence coreferent event classifier. Unlike the previous classifier, this second inter-sentence classifier places no restriction on how far apart 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 overlap with each other.

A natural question is: how can we determine whether two events are coreferent? We naively posit two events as coreferent as long as 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 it with the corresponding relation type. We could similarly create one negative training instance from every coreferent event pair that does not have any temporal relation. However, to reduce class skewness, we create negative training instances only from those coreferent event pairs where the two elements correspond to main events. Test instances are created in the same way as the negative training instances.

In our experiments, we train each of these four classifiers using SVM^{multiclass} [18]. We tune the regularization parameter, C, on the 20% of the training data that we reserved for development, and set the remaining learning parameters to their default values.[5]

4. OUR HYBRID APPROACH

In this section, we describe our hybrid learning-based and rule-based approach to temporal relation classification. Section 4.1 describes our novel features, which will be used to augment the baseline feature set (see Section 3) to train each of the four specialized classifiers mentioned above. Section 4.2 outlines our manual rule creation process. Section 4.3 discusses how we combine our hand-crafted rules and the learned classifiers.

4.1 Six Types of New Features

4.1.1 Pairwise Features

Recall that some of the features in the baseline feature set are computed based on either e_1 or e_2 but not both. Since our task is to predict the *relation* between them, we hypothesize that *pairwise* features, which are computed based on both elements, could better capture their relationship.

Specifically, we introduce pairwise versions of the head word feature and the two prepositional lexeme-based features in the baseline. In addition, we create one quadruple-wise feature by pairing up the type and modality attribute values of e_1 with those of e_2. Next, we create two *trace* features, one based on prepositions and the other on verbs, since prepositions and verb tense have been shown to play an important role in temporal relation classification. The *preposition trace* feature is computed by (1) collecting the list of prepositions along the path from e_1/e_2 to the root of its syntactic parse tree, and (2) concatenating the resulting lists computed from e_1 and e_2. The *verb trace* feature is computed in a similar manner, except that we collect the POS tags of the verbs appearing in the corresponding paths.

[5]To reduce the number of parameter tuning experiments, we find the C value that works best with the baseline classifiers and use it to train all the remaining relation classifiers in our experiments.

4.1.2 Dependency Relations

We introduce features computed based on dependency parse trees 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:

(3) It is *aggravated* by *activity*.

In (3), 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 **Simultaneous**.

Given the potential usefulness of dependency relations for temporal relation classification, we create dependency-based features as follows. For each of the 25 dependency relation types produced by the Stanford parser, we create four binary features: whether e_1/e_2 is the governing entity in the relation, and whether e_1/e_2 is the dependent in the relation.

4.1.3 Webster Relations

Some events are not connected by a dependency relation but by a *lexical* relation. We hypothesize that some lexical relations could be useful for temporal relation classification. Consider the following example.

(4) Her amylase was *mildly elevated* but has been *down* since then.

In this sentence, the two events, *mildly elevated* and *down*, are connected by an antonym relation. Statistically speaking, if (1) two events are in two clauses connected by the coordinating conjunction *but*, (2) one is an antonym of the other, and (3) there is a temporal relation between them, then not only can we infer that they do not have any temporal overlap, but it is likely that they have an asynchronous relation such as **Before** or **After**.

Given the potential usefulness of lexical relations for temporal relation classification, we create features based on four types of lexical relations present in Webster's online thesaurus[6], namely synonyms, related-words, near-antonyms, and antonyms. Specifically, for each event e appearing in the i2b2 corpus, we first use the head word of e to retrieve four lists, which are the lists corresponding to the synonyms, related words, near-antonyms, and antonyms of e. Then, given a training/test instance involving e_1 and e_2, we create eight binary features: whether e_1 appears in e_2's list of synonyms/related words/near-antonyms/antonyms, and whether e_2 appears in e_1's list of synonyms/related words/near-antonyms/antonyms.

4.1.4 WordNet Relations

Previous uses of WordNet for temporal relation classification are limited to synsets (e.g., Llorens et al. [9]). We hypothesize that other WordNet lexical relations could also be useful

[6]`http://www.merriam-webster.com/`

for the task. Specifically, we employ four types of WordNet relations, namely hypernyms, hyponyms, troponyms, and similar, to create eight binary features for temporal relation classification. These eight features are created from the four WordNet relations in the same way as the eight features were created from the four Webster relations mentioned above.

4.1.5 Predicate-Argument Relations

So far we have exploited lexical and dependency relations for temporal relation classification. We hypothesize that semantic relations, in particular predicate-argument relations, could be useful for the task. Consider the following example.

(5) 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 end/destination point. Hence, we can infer that there is a **Begins** relation between the OCCURRENCE event and the CLINICAL_DEPARTMENT event since the OCCURRENCE event begins at the end point.

Given the potential usefulness of relations between a predicate and its *numbered* arguments (e.g., A0, A1, ...) for temporal relation classification, we create one binary feature for each pairing of a numbered argument and a predicate, setting its value to 1 if according to SENNA e_1 and e_2 are in the predicate-argument relation specified by the pair.

To create additional features from predicate-argument relations, consider another PropBank-style predicate-argument relation type, *cause*. Assuming that e_2 is in e_1's cause argument, we can infer that e_2 occurs **Before** e_1, since intuitively the cause of an action precedes the action.

Consequently, we create additional features for temporal relation classification based on four types of predicate-argument relations provided by SENNA, namely directional, manner, temporal, and cause. Specifically, we create four binary features that encode whether argument e_2 is related to predicate e_1 through the four types of relations, and another four binary features that encode whether argument e_1 is related to predicate e_2 through the four types of relations.

4.1.6 Discourse Relations

Rhetorical relations such as causation, elaboration and enablement could aid in tracking the temporal progression of the discourse [7]. 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 [8] 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 two examples (see Table 2), one involving an implicit relation (Example (6)) and the other an explicit relation (Example (7)). For convenience, both sentences are also annotated using Lin et al.'s (2013) 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 (6), we aim to determine the temporal relation between two PROBLEM events, *Hypotension* and *sepsis*. The parser determines that a RESTATEMENT implicit relation exists between the two sentences. Intuitively, two temporally linked PROBLEM events within different discourse units connected by the RESTATEMENT relation implies some sort of synchronicity in their temporal relation. This means that the relation type is likely to be **Overlap** or **Simultaneous**. In this case, we can rule out **Simultaneous**: by definition, two non-coreferent events of the same type (e.g., *Hypotension* and *sepsis*) cannot have a **Simultaneous** relation.

In (7), we aim to determine the relation between the TREATMENT event *operation* and the OCCURRENCE event *benign convalescence*. The parser determines that a ASYNCHRONOUS explicit relation triggered by *thereafter* exists between the two sentences, which in turn suggests that the two events are likely to have an asynchronous temporal relation such as **Before** or **After**. By considering just the discourse connective *thereafter*, we can infer that the correct temporal relation is **Before**.

Given the potential usefulness of discourse relations for temporal relation classification, we create four features based on discourse relations. In the first feature, if e_1 is in Arg1, e_2 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 e_2 is in Arg1, e_1 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.

4.2 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 can 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.

Some of these rules were shown in Section 4.1 when we motivated each feature type with examples. Our final ruleset can be accessed via a web link (see Footnote 2).

(6)	{_Arg1 # *Hypotension*: per referral form. _Arg1} {_Arg2_RESTATEMENT Initially concern for *sepsis* in the setting of fevers and high blood count. _Arg2}
(7)	{_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}

Table 2: Examples illustrating the usefulness of discourse relations for temporal relation classification. The two arguments of each discourse relation, Arg1 and Arg2, are enclosed in curly brackets, and the sense of the relation is annotated.

4.3 Combining Rules and Machine Learning

We investigate two ways to combine the hand-crafted rules and the machine-learned classifiers.

In the first method, we employ all of the rules as additional features for training each of the four specialized classifiers. The value of each such feature is the temporal relation type predicted by the corresponding rule.

The second method can be viewed as an extension of the first one. Given a test instance, we first apply to it the ruleset composed only of rules that are at least 75% accurate. If none of the rules is applicable, we classify it using one of the four classifiers employed in the first method.[7]

5. EVALUATION: THE FIRST SETTING

5.1 Experimental Setup

In this section, we will conduct experiments under the first setting, where we assume we are given gold-standard temporal relations (i.e., each instance belongs to one of the 12 relations).

Dataset. As mentioned before, we use the 190 training documents from the i2b2 corpus for classifier training and manual rule development and reserve the 120 test documents for evaluating system performance.

Evaluation metrics. We employ *micro F-score* (F^{mi}) and *macro F-score* (F^{ma}) [see Sebastiani [16] for their definitions]. Briefly, macro F can be seen as giving the same weight to each of the 12 classes regardless of their frequency of occurrence in the test set, whereas micro F gives more weight to the frequent classes.[8] Hence, macro F could provide some insights into how well our approach performs on the minority classes.

5.2 Results and Discussion

Table 3 shows the results for our 12-class temporal relation classification task when the experiments are conducted under the first setting (see the introduction), where gold-standard temporal relations are used. The five columns of the table correspond to five different system architectures. The "Features" column corresponds to a purely learning-based system where the results are obtained simply by training a temporal relation classifier using the available features. The next two columns correspond to two purely rule-based systems, differing by whether all rules are used regardless of their accuracy or whether only high-accuracy rules (i.e., those that are at least 75% accurate) are used. The rightmost two columns correspond to the two ways of combining rules and machine learning described in Section 4.3.

On the other hand, the rows of the table differ in terms of what features are available to a system. In row 1, only the baseline features are available. In the subsequent rows, the six types of features discussed in Section 4 are added incrementally to the baseline feature set. So, the last row corresponds to the case where all feature types are used.

A point merits clarification. It may not be immediately clear how to interpret the results under, for instance, the "All Rules" column. In other words, it may not be clear what it means to add the six types of features incrementally to a rule-based system. Recall that one of our goals is to compare a purely learning-based system with a purely rule-based system, since we hypothesized that humans may be better at combining the available features to form rules than a learning algorithm. To facilitate this comparison, all and only those features that are available to a learning-based system in a given row can be used in hand-crafting the rules of the rule-based system in the same row. The other columns involving the use of rules can be interpreted similarly.

The best-performing system architecture is the hybrid architecture where high-accuracy rules are first applied and then the learned classifier is used to classify those cases that cannot be handled by the rules (see the rightmost column of Table 3). When all the features are used in combination with this architecture, the system achieves a micro F-score of 61.1% and a macro F-score of 60.7%. This translates to a relative error reduction of 15–21% in comparison to the baseline result shown in row 1. Regarding the usefulness of each of the six types of features in this best-performing architecture, we found that adding pairwise features, predicate-argument relations and discourse relations significantly improves both micro and macro F-scores.[9] The dependency, WordNet, and Webster relations are not useful.[10]

Among the remaining four architectures, the version of the rule-based system where only the high-accuracy rules are used performs the worst, owing to the low coverage of the ruleset. Comparing the "Features" system and the "All Rules" system, we see that "All Rules" is always significantly worse than "Features". These results suggest that overall, the machine learner is better at combining the available knowledge sources than the human for temporal relation

[7]The classifier that is being used for classifying a test instance depends on the test instance. For example, if the test instance is formed from two events that appear in the same sentence in the corresponding text, the intra-sentence event-event classifier will be used.

[8]Note that under the first setting, micro F is equivalent to accuracy (the percentage of correctly classified test instances), since gold-standard relations are used.

[9]All the statistical significance tests in this paper are conducted using the paired t-test ($p < 0.05$).

[10]A closer examination of the results reveals why the lexical relations extracted from WordNet and Webster are not useful. We observed that the set of verbs used to refer to events in the discharge reports (e.g., *present*, *admit*, *discharge*, *complain*) is fairly limited. This has made it comparatively easier to learn the temporal relations between the events they represent directly from the training data (e.g., a patient has to be *admitted* first before being *discharged*), rendering the WordNet and Webster relations less useful.

		Features		All Rules		All Rules with accuracy ≥ 0.75		Features + Rules as Features		Rules + Features + Rules as Features	
	Feature Type	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}
1	Baseline	55.3	50.5	–	–	–	–	–	–	–	–
2	+ Pairwise	55.5	51.1	37.6	25.0	14.5	17.6	57.2	52.8	57.8	52.4
3	+ Dependencies	55.5	51.2	40.0	31.9	16.2	23.3	57.4	53.1	58.1	53.0
4	+ WordNet	55.6	51.1	40.0	31.9	16.2	23.3	57.2	53.0	57.9	52.9
5	+ Webster	55.8	51.3	40.0	31.9	16.2	23.3	57.3	53.0	58.0	52.9
6	+ PropBank	55.8	51.3	45.4	44.7	21.3	34.7	57.6	53.1	59.7	57.7
7	+ Discourse	56.2	51.5	47.3	47.8	24.0	39.2	57.9	53.2	**61.1**	**60.7**

Table 3: Micro and macro F-scores of classifying gold-standard temporal relations as features are added incrementally to the baseline.

classification. The question, however, is: does the machine learner make mistakes on different instances than the human? By comparing the results of the two feature-based systems, "Features" and "Features + Rules as Features", we can infer that the answer is yes. Since the latter is significantly better than the former, the incorporation of the hand-crafted rules into the feature set is beneficial for the learner. In other words, the use of rules as features helps fix some of the mistakes made by the learner.

6. TEMPORAL RELATION IDENTIFICATION

In the previous section, we evaluated our approach under the first setting, where we assume we are given only instances that belong to one of the 12 relation types. Recall from the introduction that we make the task more challenging by also evaluating our approach under a second setting, where we assume the instances we are given may or may not belong to one of the 12 relation types. For the second setting, we adopt a *pipeline* system architecture where we first employ a relation *identification* system to determine whether a test instance possesses a temporal relation. We then use the relation *classification* system described in Section 4 to classify only those instances the relation identification system determined possessed a temporal relation. The rest of this section describes our temporal relation *identification* system.

Given the success of our hybrid approach to relation classification, we employ a hybrid approach to relation identification. Specifically, given a test instance i, we first apply a set of hand-crafted rules to determine whether i has a relation. If i cannot be classified by any of the rules, we employ a learned identifier to determine whether i has a relation.

Two questions naturally arise. First, how can we design the hand-crafted rules? Second, how can we train a classifier for identifying relations? We answer these two questions in the next two paragraphs.

We design the hand-crafted identification rules as follows. As *positive* rules (i.e., rules that determine that an instance has a relation), we simply use all the rules that we hand-crafted for relation classification in Section 4.2. To design *negative* rules (i.e., rules that determine that an instance has no relation), we employ the same data-driven procedure that was used to design the relation classification rules (see Section 4.2).

Next, we describe how to train a classifier for identifying temporal relations. We employ a natural way of creating training instances: we use all event-event and event-time pairs in the training set that have a relation as positive instances, and the remaining ones as negative instances. As before, rather than training just one classifier for identifying temporal relations, we train four specialized classifiers for identifying relations using the same division that we described in Section 3.1. It is worth mentioning, however, that the negative instances significantly outnumber the positive ones, since most pairs do not have a relation. But since training on a dataset with a skewed class distribution may adversely affect the performance of a classifier, for each of the four specialized classifiers, we employ simple pruning heuristics to prune the negative training instances before training the classifier.[11]

The remaining question is: what features should we use to represent each training/test instance? We experimented with three options. The simplest option is to employ the same features that we used to train our classifiers for relation classification in Section 4. Note that many of these features are extracted from syntactic parse trees. Since it is not clear whether these features have adequately encoded all the useful information that we can possibly extract from a parse tree, perhaps the simpler thing to do, which we consider in our second option, is to employ just the syntactic parse tree containing the two entities involved in an instance.[12] Recall that advanced machine learning algorithms such as SVMs have enabled a parse tree to be used as a *structured* feature (i.e., a feature whose value is a linear or hierarchical structure, as opposed to a *flat* feature, which has a discrete or real value), owing to their ability to employ *kernels* to efficiently compute the similarity between two potentially complex structures. In particular, given two parse trees, we compute their similarity using a convolution tree kernel [4].

Note, however, that while we want to use a parse tree directly as a feature, we do *not* want to use the *entire* parse tree as a feature. Specifically, while using the entire parse tree enables a richer representation of the syntactic context of the two entities than using a *partial* parse tree, the increased complexity of the tree also makes it more difficult for the SVM learner to make generalizations.

To strike a better balance between having a rich representation of the context and improving the learner's ability to generalize, we extract a subtree from a parse tree and use it as the value of the structured feature of an instance. Specifically, given two entities in an instance and the associated

[11] For the complete list of pruning heuristics, see `http://www.hlt.utdallas.edu/~jld082000/temporal-relations/`.

[12] If the two entities involved appear in different sentences, we create a parse tree by connecting the roots of the two parse trees in which the two entities appear to a pseudo root node.

syntactic parse tree T, we retain as our subtree the portion of T that covers (1) all the nodes lying on the shortest path between the two entities, and (2) all the immediate children of these nodes that are not the leaves of T.

This subtree is known as a *simple expansion tree*, and was first used by Yang et al. [22] as a structured feature for the pronoun resolution task. Note that some of the flat features employed in the first option, including the event attributes (i.e., type, polarity and modality) and the time attribute (i.e., type), are not encoded in the simple expansion tree. Hence, we encode these attribute values in the tree as follows: we replace the parent node of each entity under consideration with its event/time attribute values. To better understand how a simple expansion tree is computed, we show in Figure 1 the simple expansion tree created for the events *evaluated* and *desaturate*. Note that only those nodes that are circled or squared are part of the tree.

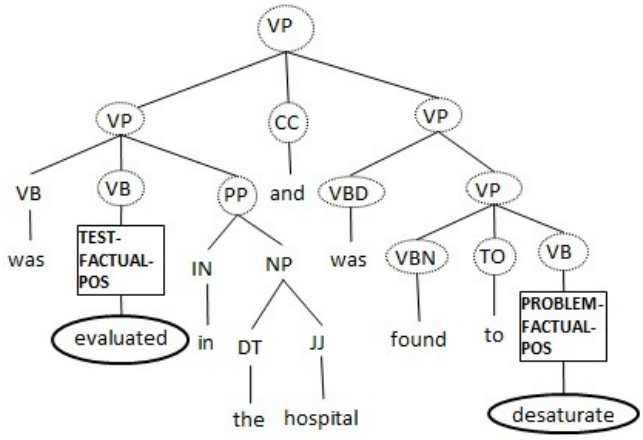

Figure 1: Example of a simple expansion tree.

Given that we employ flat features in our first option and a tree feature in our second option, a natural third option is to combine the flat and tree features to train a classifier. To compute the similarity between two instances containing both flat and tree features, we first compute the similarity of their flat features using a linear kernel and the similarity of their tree features using a tree kernel, and then combine these two kernels using a composite kernel.[13]

After training the four specialized classifiers, we can apply them to classify whether a test instance has a relation or not. By default, any instance whose classification value is at least 0 is classified as having a relation; otherwise, it is classified as having no relation.[14]

Since we are using the relation *identification* system to filter the no relation instances prior to relation *classification*, the performance of the downstream relation classification system depends to a large extent on the performance of the identification system. If the identification system misclassifies many positive instances (as negative), it will harm the recall of the classification system; on the other hand, if it misclassifies many negative instances (as positive), it will harm the precision of the classification system.

[13]In preliminary experiments on the development data, the second option yields marginally better results than the others. So the results we report in the next section are based on identification classifiers trained using the second option.

[14]The classification value of an instance is simply its signed distance from the SVM hyperplane.

Ideally, we want to optimize the performance of the identification classifier such that when it is used in combination with the *classification system*, the F-measure of the classification system is maximized. However, the identification classifier is trained to maximize classification accuracy on identification. To maximize the F-measure of the classification system instead, we propose to adjust the *classification threshold* (i.e., the threshold that determines whether an instance should be classified as positive or not). Recall that currently we employ a classification threshold of 0, meaning that all and only those instances whose classification value is 0 or above are classified as positive. By adjusting this threshold, we can potentially vary the F-measure of the classification system. Specifically, by lowering the threshold, more instances will be classified as positive, potentially improving the recall of the classification system. By the same token, increasing the threshold could improve its precision.

Given this observation, we tune the classification threshold to maximize the F-measure of the classification system on the development set, which is composed of 20% of the training data. In other words, we first train both the classifiers for relation identification and classification on the remaining 80% of the training data. Then we obtain relation classification results on the development set by varying the classification thresholds applied to the relation identification classifiers (each of the four specialized identification classifiers will have its classification threshold tuned independently of the others).[15] The thresholds that yield the best relation classification F-measure score on the development set are applied to obtain relation classification results on the test data. Since our results are reported in terms of both micro and macro F-scores, we obtain thresholds that maximize macro F and those that maximize micro F separately.

7. EVALUATION: THE SECOND SETTING

Next, we conduct experiments under the second setting, where we obtain temporal relation classification results using automatically identified temporal relations.

Results, expressed in terms of micro and macro F, are shown in Table 4, where the rows and columns can be interpreted in the same manner as those in Table 3. As expected, the results obtained using automatically identified relations are significantly lower than those obtained using gold-standard temporal relations. Nevertheless, the same conclusions that we drew from the results in Table 3 are also applicable to the results in Table 4. It is worth mentioning, however, that the best-performing system is still the "Rules + Features + Rules as Features" architecture when used in combination with all the feature types, achieving a micro F-score of 30.0% and a macro F-score of 38.8%. This translates to a significant relative error reduction of 6–13% in comparison to the baseline.

8. ERROR ANALYSIS

To gain additional insights into the errors made by the relation classification system and the relation identification system, we perform an error analysis of each of them.

8.1 Relation Classification Errors

[15]We attempted thresholds from -1.0 to 1.0 in steps of 0.1.

		Features		All Rules		All Rules with accuracy ≥ 0.75		Features + Rules as Features		Rules + Features + Rules as Features	
	Feature Type	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}
1	Baseline	26.0	29.7	–	–	–	–	–	–	–	–
2	+ Pairwise	26.5	30.5	14.8	12.3	6.8	7.8	26.6	31.0	27.5	32.3
3	+ Dependencies	26.5	30.6	17.1	18.9	9.9	14.3	26.7	31.4	27.7	33.0
4	+ WordNet	26.5	30.7	17.2	18.9	9.9	14.3	26.6	31.2	27.6	32.9
5	+ Webster	26.5	30.7	17.2	19.0	9.9	14.3	26.7	31.2	27.6	32.9
6	+ PropBank	26.5	30.8	21.2	29.3	15.4	24.4	26.8	31.3	29.1	36.7
7	+ Discourse	26.6	30.8	21.9	30.5	18.8	29.6	26.8	31.3	**30.0**	**38.8**

Table 4: Micro and macro F-scores of classifying automatically identified temporal relations as features are added incrementally to the baseline.

We constructed the confusion matrix based on the gold-standard and predicted relation types on the test set, and found that there are three types of confusions that account for nearly 72% of the classification errors. Below we illustrate each of these three types of confusions with examples.

Simultaneous confused as Overlap. This is the most frequent source of confusion, accounting for 30.8% of the errors. The following example illustrates this confusion:

04-24 PICC Bld Cx : pseudomonas Diaz to zosyn , cipro , cefepime , Tardugno – staph epi- Gray to vanc

In this sentence, the treatments 'zosyn', 'ciporo', 'cefepime', and 'Tardugno' are all given at the same time, and therefore are temporally **Simultaneous**. However, there are many cases where events separated by commas are overlapping rather than simultaneous. Determining whether the relation should be **Simultaneous** or **Overlap** requires an understanding of the nature of the events and cannot simply be inferred based on syntactic patterns. This poses a challenge to the relation identification system.

Recall that the i2b2 organizers grouped **Overlap** and **Simultaneous** under the same broad relation type. The fact that almost a third of our relation classification errors were related to confusion between **Overlap** and **Simultaneous** seems to be consistent with the notion that merging them was a wise decision. As Pustejovsky and Stubbs [15] point out, categorization results may lead a human annotator to re-think her annotation model. In this case, our error analysis seems to support the redesigned model (i.e., with **Overlap** and **Simultaneous** combined).

Before confused as Overlap. This is the second most frequent source of confusion, accounting for 21.5% of the errors. The following example illustrates this confusion:

She [called 911] and he was [brought] to Hahnemann General Hospital Lydia.

In this sentence, OCCURRENCE event *called 911* is temporally **Before** OCCURRENCE event *brought*, but the relation is misclassified as **Overlap**. This source of confusion arises from the presence of the co-ordinating conjunction "and", which frequently appears together with the **Overlap** relation. In this example, understanding that *called 911* took place before *bought* requires world knowledge, which might be acquired via narrative chains [1].

After confused as Overlap. This is the third most frequent source of confusion, accounting for 19.5% of the errors. The following example illustrates this confusion.

Also , a repeat outpatient [CT colonoscopy] with [better preparation] should be considered.

In this sentence, TEST event *CT colonoscopy* is proposed **After** OCCURRENCE event *better preparation* in the gold standard, but the relation is misclassified as **Overlap**. The difficulty in correctly classifying this relation as **After** arises from the fact that an OCCURRENCE event can be anything that is clinically relevant to the patient's timeline apart from the other defined attributes, and hence it can take on various temporal roles depending on whether it is in an adverbial phrase, an adjectival phrase, a noun phrase, or a verb phrase.

Intuitively, when an event happens 'with' another event, they generally tend to have temporal synchronicity, and in such cases entity attribute information may not be so important. However, if there isn't temporal synchronicity (as in the above sentence), then we will need to rely on information reflected by entity attributes, especially type. More specifically, to classify the relation type correctly, we will need to narrow the scope of OCCURRENCE events by including more event types that are clinically relevant to the current set of event types. These event types might include CONDITION instead of OCCURRENCE for phrases such as 'doing well', 'improving', etc., or PREP for events that are not TREATMENTS but are necessary as a step before the TREATMENT, as in the above sentence.

8.2 Relation Identification Errors

For relation identification, we will perform a qualitative analysis, since it is harder to perform the kind of quantitative analysis that we did for relation classification.

One common source of errors involves cases whose relation type may be difficult even for humans to determine. Consider, for example, events listed as a sequence, as shown in the sentence below:

[CXR] , [LP] , [UA] and [abdominal CT] showed no sign of infection.

Here, the sequence of TESTS paired consecutively as CXR and LP, LP and UA, and UA and abdominal CT are unannotated in the dataset with a relation, but the identification classifier classifies them as having a relation.

Note that for sentences like this where the patient's past history of problems is listed, it can sometimes be difficult even for a human to determine the exact temporal relation type between the events, as a mixture of temporal relations such as **Overlap**, **Before**, **After**, etc. can exist. When a case appears temporally undeterministic to a human annotator, she may choose to leave them unannotated. In other

words, even though these cases are counted as errors made by our identification system, they probably shouldn't be.

Another common source of errors involves coreferent events that appear in different sentences. Recall that we naively posit two events that have the same head as coreferent, and train a classifier to determine whether there is a relation between two such events. We noticed that this classifier classifies all instances as having a relation. However, there are many same-head events that do not have a temporal relation. To address this problem, we will need to employ a coreference classifier to determine whether two same-head events are coreferent.

The third common source of errors stems from the fact that not all temporal relations in the dataset are annotated. Consider the two sentences below:

ESRD on HD - Pt has [ESRD] secondary to [her DM] and is on HD.
Pt is now [transferred] to the FICU for [further care].

In the first sentence, the PROBLEM event pair *ESRD* and *her DM*, which should have relation type **Overlap_After**, are not annotated as having a relation in the dataset, but our identification classifier determines that it does. In the second sentence, the OCCURRENCE event *transferred* and the TREATMENT event *further care*, which should have relation type **Before**, is not annotated as having a relation in the dataset, but our identification classifier determines that they do. As in the first type of errors discussed above, even though these cases are counted as errors made by our identification system, they probably shouldn't be.

Overall, this qualitative analysis reveals that the error rate of our relation identification system is to some extent inflated owing to the incompleteness of the gold-standard annotations. While the performance of our relation classification system significantly degrades when gold-standard temporal relations are replaced by their automatically identified counterparts, we speculate that the degradation will not be as abrupt as what we currently see given a better-prepared set of gold-standard annotations.

9. CONCLUSIONS

We have investigated a knowledge-rich, hybrid approach to the 12-class temporal relation classification task for the clinical domain. Results on the i2b2 corpus shows that when evaluated on gold-standard and automatically identified temporal relations, our approach achieves a relative error reduction of 15–21% and 6–13% respectively over a state-of-the-art learning-based baseline.

10. ACKNOWLEDGMENTS

We thank the three reviewers for their invaluable comments on an earlier draft of the paper. This work was supported in part by NSF Grants IIS-1147644 and IIS-1219142.

11. REFERENCES

[1] N. Chambers and D. Jurafsky. Unsupervised learning of narrative event chains. In *Proceedings of ACL-HLT*, pages 787–797, 2008.

[2] N. Chambers, S. Wang, and D. Jurafsky. Classifying temporal relations between events. In *Proceedings of the ACL Companion Volume*, pages 173–176, 2007.

[3] T. Chklovski and P. Pantel. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of EMNLP*, pages 33–40, 2004.

[4] M. Collins and N. Duffy. Convolution kernels for natural language. In *Proceedings of NIPS*, pages 625–632, 2001.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[6] E. Y. Ha, A. Baikadi, C. Licata, and J. Lester. NCSU: Modeling temporal relations with markov logic and lexical ontology. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 341–344, 2010.

[7] J. Hitzeman, M. Moens, and C. Grover. Algorithms for analysing the temporal structure of discourse. In *Proceedings of EACL*, pages 253Ű-260, 1995.

[8] Z. Lin, H. T. Ng, and M.-Y. Kan. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering (to appear)*, 2013.

[9] H. Llorens, E. Saquete, and B. Navarro. 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, 2010.

[10] I. Mani, M. Verhagen, B. Wellner, C. M. Lee, and J. Pustejovsky. Machine learning of temporal relations. In *Proceedings of COLING/ACL*, pages 753–760, 2006.

[11] C. Min, M. Srikanth, and A. Fowler. LCC-TE: A hybrid approach to temporal relation identification in news text. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pages 219–222, 2007.

[12] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. The Penn Discourse Treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, 2008.

[13] G. Puşcaşu. WVALI: Temporal relation identification by syntactico-semantic analysis. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pages 484–487, 2007.

[14] J. Pustejovsky, P. Hanks, R. Sauri, A. See, D. Day, L. Ferro, R. Gaizauskas, M. Lazo, A. Setzer, and B. Sundheim. The TimeBank corpus. In *Corpus Linguistics*, pages 647–656, 2003.

[15] J. Pustejovsky and A. Stubbs. *Natural Language Annotation and Machine Learning*. O'Reilly Publishers, 2012.

[16] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[17] B. Tang, Y. Wu, M. Jiang, Y. Chen, J. Denny, and H. Xu. Extracting temporal information from clinical text — Vanderbilt's system for the 2012 i2b2 NLP challenge. In *Proceedings of the 6th i2b2 Shared Tasks and Workshop on Challenges in Natural Langauge Processing for Clincial Data*, 2012.

[18] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of ICML*, pages 104–112, 2004.

[19] M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. SemEval-2007 Task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations*, pages 75–80, 2007.

[20] M. Verhagen, R. Sauri, T. Caselli, and J. Pustejovsky. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, 2010.

[21] Y. Xu, Y. Wang, T. Liu, J. Tsujii, and E. Chang. An end-to-end system to identify temporal relation in discharge summaries. In *Proceedings of the 6th i2b2 Shared Tasks and Workshop on Challenges in Natural Langauge Processing for Clincial Data*, 2012.

[22] X. Yang, J. Su, and C. L. Tan. Kernel based pronoun resolution with structured syntactic knowledge. In *Proceedings of COLING/ACL*, pages 41–48, 2006.