Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: (NAACL HLT), Atlanta, GA, June 2013, pp. 918--927.

Classifying Temporal Relations with Rich Linguistic Knowledge

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. Unlike existing approaches to this task, we (1) classify an event-event or event-time pair as one of the 14 temporal relations defined in the TimeBank corpus, rather than as one of the six relations collapsed from the original 14; (2) employ sophisticated linguistic knowledge derived from a variety of semantic and discourse relations, rather than focusing on morpho-syntactic knowledge; and (3) leverage a novel combination of rule-based and learning-based approaches, rather than relying solely on one or the other. Experiments with the TimeBank corpus demonstrate that our knowledge-rich, hybrid approach yields a 15–16% relative reduction in error over a state-of-the-art learning-based baseline system.

1 Introduction

Recent years have seen a surge of interest in temporal information extraction (IE). Temporal relation classification, one of the most important temporal IE tasks, involves classifying a given event-event pair or event-time pair as one of a set of predefined temporal relations. The creation of the TimeBank corpus (Pustejovsky et al., 2003) and the organization of the TempEval-1 (Verhagen et al., 2007) and TempEval-2 (Verhagen et al., 2010) evaluation exercises have facilitated the development and evaluation of temporal relation classification systems.

Our goal in this paper is to advance the state of the art in temporal relation classification. 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 six temporal relations as is typically done in previous work (see Section 2 for more information), we address an arguably more challenging version of the task where we consider all the 14 relations originally defined in the TimeBank corpus.

Our approach to temporal relation classification can be distinguished from existing approaches 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 relied primarily on morpho-syntactic features as well as a few semantic features extracted from WordNet synsets and VerbOcean's (Chklovski and Pantel, 2004) 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) semantic features encoding a variety of semantic relations, including PropBank-style predicate-argument relations as well as those extracted from the Merriam-Webster dictionary, and (2) discourse features encoding automatically computed Penn Discourse TreeBank (PDTB) style (Prasad et al., 2008) discourse relations.

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 two hypotheses. First, a rule-based method could better handle the skewed class distribution underlying the dataset for our 14-class classification problem. 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 shown to underperform learning-based approaches on this task (Mani et al., 2006), 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., Mani et al. (2006), Puşcaşu (2007)), our rules are created in a *data-driven* manner via a manual inspection of the annotated temporal relations in the TimeBank corpus.

Experiments on the TimeBank corpus demonstrate the effectiveness of our knowledge-rich, hybrid approach to temporal relation classification: it yields a 15–16% relative reduction in error over a state-of-the-art learning-based baseline system.

To our knowledge, we are the first to (1) report results for the 14-class temporal relation classification task on the TimeBank (v1.2) corpus; (2) successfully employ automatically computed 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. Note that hybrid approaches in this spirit were popular in the natural language processing community back in the mid-90s (Klavans and Resnik, 1994). We believe that they are among the most competitive approaches to language processing tasks that require complex reasoning and should be given more attention in the community. We release the complete set of rules that we mined from the TimeBank corpus and used in our rule-based approach in hopes that our insights into how features can be combined as decision rules can benefit researchers interested in this task.

The rest of the paper is organized as follows. Section 2 provides an overview of the TimeBank corpus. Sections 3 and 4 describe the baseline system and our approach, respectively. We present evaluation results in Section 5 and conclude in Section 6.

2 Corpus

For evaluation, we use the TimeBank (v1.2) corpus, which consists of 183 newswire articles. In each article, the *events*, *times*, and their *temporal relations* are marked up. An event, which can be a tensed verb, adjective, or nominal, contains various attributes, including the *class* of event, *tense*, *aspect*, *polarity*, and *modality*. A time expression has a *class* attribute, which specifies whether it is a date, time, duration, or set, and its value is normalized based on TIMEX3. A temporal relation can be an *order* relation, which orders two events (as in sentence (1)), or an *anchor* relation, which anchors an event to a time expression (as in sentence (2)).

(1) A steep *rise* in world oil prices followed the Kuwait *invasion*.

(2) We are there to *stay* for a long *period*.

Each temporal relation has a *type*. For example, the relation defined on *rise* and *invasion* in (1) has type **After**, whereas the relation defined on *stay* and *period* in (2) has type **During**. Note that a temporal relation is defined on an *ordered* pair. For example, in (1), the pair (*rise*, *invasion*) has type **After**, whereas the pair (*invasion*, *rise*) has type **Before**).

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

In our experiments, we assume that our temporal relation classification system is given an event-event or event-time pair that is known to belong to one of the 14 relation types defined in TimeBank and aims to determine its relation type. Following previous evaluations of the temporal relation classification task on the TimeBank corpus (e.g., Mani et al. (2006), Chambers et al. (2007)) and in TempEval-1/2, we assume as input gold events and time expressions.

Unlike Mani et al. (2006) and Chambers et al. (2007), who focus on six relation types (**Simultaneous**, **Before**, **IBefore**, **Begins**, **Ends**, and **Includes**), we report results on 14 relation types. Note that the aforementioned six relation types are chosen by (1) discarding **During**, **During_Inv**, and **Identity**, and (2) combining the two relation types in each of the five pairs, namely (**Before**, **After**), (**IBefore**, **IAfter**), (**Includes**, **Is_Included**), (**Begins**, **Begun_By**), and (**Ends**, **Ended_By**), into a single type because they are inverses of each other. In other words, if a relation instance (e_1, e_2) is anno-

Id	Relation	Description	Total %	E-E	E-T
1	**Simultaneous**	e_1 and e_2 happen at the same time or are temporally distinguishable	660 (13.3)	599	61
2	**Identity**	e_1 and e_2 are coreferent	702 (14.1)	696	6
3	**Before**	e_1 happens before e_2 in time	689 (13.9)	639	50
4	**After**	e_1 happens after e_2 in time	744 (15)	681	63
5	**IBefore**	e_1 happens immediately before e_2 in time	39 (0.8)	38	1
6	**IAfter**	e_1 happens immediately after e_2 in time	28 (0.6)	25	3
7	**Includes**	As in *Ed arrived in Seoul last Sunday* (e_1=*last Sunday*; e_2=*arrived*)	758 (15.3)	318	440
8	**Is_Included**	As in *Ed arrived in Seoul last Sunday* (e_1=*arrived*; e_2=*last Sunday*)	762 (15.3)	201	561
9	**During**	e_1 persists throughout duration e_2	102 (2.1)	19	83
10	**During_Inv**	e_2 persists throughout duration e_1	124 (2.5)	44	80
11	**Begins**	e_1 marks the beginning of e_2	66 (1.3)	44	22
12	**Begun_By**	e_2 marks the beginning of e_1	61 (1.2)	32	29
13	**Ends**	e_1 marks the end of e_2	66 (1.3)	21	45
14	**Ended_By**	e_2 marks the end of e_1	170 (3.42)	93	77

Table 1: The 14 temporal relations and their frequency of occurrences in TimeBank (v1.2). Each relation is defined on an ordered event-event or event-time pair (e_1,e_2). The "Total" and "%" columns show the number and percentage of instances annotated with the corresponding relation 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.

tated as **After**, it is replaced with the instance (e_2, e_1) with class **Before**, and subsequently a relation classifier is presented with (e_2, e_1) but not (e_1, e_2). On the other hand, our 14-class task is arguably more challenging since our system has to further distinguish a relation type from its inverse given an instance in which the two elements are in arbitrary order.

3 Baseline Temporal Relation Classifier

Since the currently best-performing systems for temporal relation classification 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 14 TimeBank 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 68 linguistic features modeled after the top-performing temporal relation classification systems on TimeBank (e.g., Mani et al. (2006), Chambers et al. (2007)) and in the TempEval shared tasks (e.g., Min et al. (2007), Puşcaşu (2007), Ha et al. (2010), Llorens et al. (2010), Mirroshandel and Ghassem-Sani (2011)).[1] These features can be divided into six categories, as described below.

Lexical (5). The strings of e_1 and e_2, the head words of e_1 and e_2, and a binary feature indicating 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 word of e_1 and its preceding token, the POS bigram formed from the head word of 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 is headed by a PP (Chambers et al., 2007), the prepositional lexeme of the PP if e_2 is headed by a PP, the prepositional lexeme of the PP if e_1 is governed by a PP (Mirroshandel and Ghassem-Sani, 2011), the prepositional lexeme of the PP if e_2 is governed by a PP, the POS of the head of the verb phrase (VP) if e_1 is governed by a VP, the POS of the head of the VP if 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.[2]

[1] Note, however, that these features were designed for the arguably simpler 6-class temporal relation classification tasks.

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

Entity attributes (13). The tense, aspect, modality, polarity, and event type of e_1 and e_2 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 tense values, the aspect values, and the class values of e_1 and e_2.

Semantic (7). The subordinating temporal role token of e_1 if it appears within a temporal semantic role argument (Llorens et al., 2010), the subordinating temporal role token of e_2 if it appears within a temporal semantic role argument, the first WordNet synset to which e_1 belongs, the first WordNet synset to which e_2 belongs, and whether e_1 and e_2 are in the *happens-before*, *happens-after*, and *similar* relation according to VerbOcean.[3]

Distance (1). Are e_1 and e_2 in the same sentence?

DCT related (3). The temporal relation type between e_1 and the document creation time (DCT) [its value can be one of the 14 relation types, or NULL if no relation exists], the temporal relation type between e_2 and the DCT, and whether e_1 and e_2 have different relation types with the DCT.

After creating the training instances, we train a 14-class classifier on them using SVM^{multiclass} (Tsochantaridis et al., 2004).[4] We then use it to make predictions on the test instances, which are generated in the same way as the training instances.

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 a temporal relation classifier. Section 4.2 outlines our manual rule creation process. Section 4.3 discusses how we combine our hand-crafted rules and the learned classifier to make predictions in our hybrid approach.

[3] *happens-after* is not a relation in VerbOcean: we create this relation simply by inverting the *happens-before* relation.

[4] For all the experiments involving SVM^{multiclass}, we set C, the regularization parameter, to 10,000, since preliminary experiments indicate that preferring generalization to overfitting (by setting C to a small value) tends to yield poorer classification performance. The remaining learning parameters are set to their default values.

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 the relationship between them.

Specifically, we introduce pairwise versions of the head word feature and the two prepositional lexeme-based features in the baseline. In addition, we create two quadruple-wise features, one by pairing up the tense and class attribute values of e_1 with those of e_2, and the other by pairing up their tense and aspect values. Next, we create two *trace* features, one based on prepositions and the other on verbs, since prepositions and verb tenses 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 trees, 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.

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) Ed *changed* his plans as the mood *took* him.

In (3), there is a adverbial clause modifier dependency between *changed* and *took*, because *took* appears in an adverbial clause (headed by *as*) modifying *changed*. Intuitively, if the two events participate in this type of dependency relation and the adverbial clause is headed by *as* and there is a temporal relation between them, then it is likely that this temporal relation is **Simultaneous**. While the temporal relation type is dependent on the connective heading the adverbial clause, in general an adverbial clause modifier dependency between two events implies that their temporal relation is likely to be **Si-**

multaneous, **Before**, or **After**.

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 of these lexical relations could be useful for temporal relation classification. Consider the following example.

(4) The phony war has *finished* and the real referendum campaign has *begun*.

In this sentence, the two events, *finished* and *begun*, are connected by an antonym relation. Statistically speaking, if (1) two events are in two clauses connected by a coordinating conjunction (e.g., *and*), (2) one is an antonym of the other, and (3) there is a temporal relation between them, then the temporal relation is likely to be **Simultaneous**.

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[5], namely synonyms, related-words, near-antonyms, and antonyms. Specifically, for each event e appearing in TimeBank, 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. (2010)). We hypothesize that other WordNet lexical relations could also be useful 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 in the previous subsection.

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) "What sector is *stepping forward* to pick up the slack?" he asked.

Using SENNA (Collobert et al., 2011), a PropBank-style semantic role labeler, we know that *forward* is in the directional argument of the predicate *stepping*. This enables us to infer that an **Includes** relation exists between *stepping* and *forward* since intuitively an action includes a direction.

As another example, consider another PropBank-style predicate-argument relation, *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 features for temporal relation classification based on four types of PropBank-style predicate-argument relations, namely directional, manner, temporal, and cause. Specifically, using SENNA's output, we create four binary features that encode whether argument e_2 is related to predicate e_1 through the four types of relations, and we create 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 (Hitzeman et al., 1995). 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 clas-

[5] http://www.merriam-webster.com/

(6)	{_Arg1 Hewlett-Packard Co. *said* it raised its stake in Octel Communications Corp. to 8.5% of the common shares outstanding. _Arg1} {_Arg2_RESTATEMENT In a Securities and Exchange Commission *filing*, Hewlett-Packard said it now holds 1,384,119 Octel common shares _Arg2}.
(7)	{_Arg1 Reports *said* that Saudi Arabia told U.S. oil companies of a 15–20 percent cutback in its oil supply in September. _Arg1} {_Conn SYNCHRONY Meanwhile _Conn} {_Arg2 Egypt's Middle East Agency said *Thursday* that Saddam was the target of an assassination attempt. _Arg2}

Table 2: Examples illustrating the usefulness of discourse relations for temporal relation classification.

sification. 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 (2013) 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 the two arguments of the relation and *sense* is the sense/type of the relation. 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 order relation between the reporting event *said* and the occurrence event *filing*. The parser determines that a RESTATEMENT implicit relation exists between the two sentences. Intuitively, if no asynchronous relations can be found among the events in two discourse units connected by the RESTATEMENT relation, then the temporal relation between two temporally linked events within these units is likely to be either **Identity** or **Simultaneous**. In this case, we can rule out **Identity**: since *said* and *filing* belong to different event classes, they are not coreferent.

In (7), we aim to determine the anchor relation between the reporting event *said* and the date *Thursday*. The parser determines that a SYNCHRONY explicit relation triggered by *Meanwhile* exists between the two sentences. Intuitively, if a temporally related reporting event and date occur within different discourse units connected by the SYNCHRONY relation, then it is likely that the event **Is_Included** in the date. Note that without this discourse relation, it could be difficult for a machine to confidently associate a reporting event with a date occurring in a different discourse segment.

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 a 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 also 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 linguistic 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 TimeBank data. For this purpose, we partition the TimeBank documents into five folds of roughly the same size, reserving three folds for developing our rules and using the remaining two folds 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 three development folds. A new instance is classified using the first applicable rule in the ruleset.

Some of these rules were shown in the previous subsection when we motivated each feature type with examples. The complete set of rules can be accessed via our website.[6]

4.3 Combining Rules and Machine Learning

We investigate three 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 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 80% accurate. If none of the rules is applicable, we classify it using the classifier employed in the first method.[7]

The third method is essentially the same as the second, except we do not employ the rules as features when training the classifier.

5 Evaluation

5.1 Experimental Setup

Dataset. As mentioned before, we partition the 183 documents in the TimeBank (v1.2) corpus into five folds of roughly the same size, reserving three folds (say Folds 1–3) for manual rule development and using the remaining two folds (say Folds 4–5) for testing. We perform two-fold cross-validation experiments using the two test folds. In the first fold experiment, we train a temporal relation classifier on Folds 1–4 and test on Fold 5; and in the second fold experiment, we train the classifier on all but Fold 4 and test on Fold 4. The results reported in the rest of the paper are averaged over the two test folds.

Evaluation metrics. We employ *accuracy* (Acc) and *macro F-score* (F^{ma}). Accuracy is the percentage of correctly classified test instances, and is the standard evaluation metric for temporal relation classification. Since each test instance belongs to one of the 14 temporal relation types, accuracy is the same as micro F-score. On the other hand, macro F-score is rarely used to evaluate this task. We chose it because it could provide insights into how well our approach performs on the minority classes.

5.2 Results and Discussion

Table 3 shows the two-fold cross-validation results for our 14-class temporal relation classification task. The six columns of the table correspond to six different system architectures. The "Feature" column corresponds to a purely learning-based architecture 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 architectures, differing by whether all rules are used regardless of their accuracy or whether only high-accuracy rules (i.e., those that are at least 80% accurate) are used. The rightmost three columns correspond to the three 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. This means that 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-

[6]http://www.hlt.utdallas.edu/~jld082000/temporal-relations/

[7]Although this classifier is applied to only those test instances that the rules cannot handle, we did not retrain it on only those training instances that the rules cannot handle.

		Features		All Rules		All Rules with accuracy ≥ 0.8		Features + Rules as Features		Rules + Features		Rules + Features + Rules as Features	
	Feature Type	Acc	F^{ma}	Acc	F^{ma}	Acc	F^{ma}	Acc	F^{ma}	Acc	F^{ma}	Acc	F^{ma}
1	Baseline	45.3	24.9	–	–	–	–	–	–	–	–	–	–
2	+ Pairwise	46.5	25.8	37.6	26.5	5.1	13.9	46.7	26.5	48.0	31.9	48.2	32.1
3	+ Dependencies	47.0	25.9	39.0	27.8	6.9	15.7	47.2	26.7	49.2	32.3	49.2	32.6
4	+ WordNet	46.9	26.0	43.5	30.4	6.9	15.7	47.5	26.8	49.2	32.3	49.5	32.8
5	+ Webster	46.9	25.8	43.3	29.9	6.9	15.7	48.1	26.8	49.2	32.0	50.1	33.1
6	+ PropBank	47.2	26.0	44.3	30.5	8.1	16.6	48.0	26.8	49.5	32.2	50.0	33.0
7	+ Discourse	48.1	26.6	47.5	35.1	12.8	23.3	48.9	27.5	53.0	36.0	**53.4**	**36.6**

Table 3: Two-fold cross-validation accuracies and macro F-scores as features are added incrementally to the baseline.

based system, since we hypothesized that humans may be better at combining the available features to form rules than a learning algorithm would be. 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 in a similar manner.

The highest accuracy and macro F-score are achieved when all types of features are used in combination with the "Rules + Features + Rules as Features" architecture. Specifically, this system achieves an accuracy of 53.4% and a macro F-score of 36.6% on the 2000-instance test set. This translates to a relative error reduction of 15–16% in comparison to the baseline result shown in row 1. A closer examination of these results reveals that the hand-crafted rules used by the system correctly classify 239 of the 305 test instances to which they are applicable. In other words, the rules achieve a precision of 78.3% and a recall of 15.3% on the test data.

Our results suggest that the rules are effective at improving performance when they are used to make classification decisions prior to the application of the classifier, as the performance of the "Rules + Features + Rules as Features" architecture is significantly better than that of the "Features + Rules as Features" architecture.[8] On the other hand, the "Rules + Features + Rules as Features" architecture does not benefit from the use of rules as features, as its performance is statistically indistinguishable from that of the "Rules + Features" architecture. Nevertheless, both "Rules + Features + Rules as Features" and "Rules + Features" are significantly better than the remaining four architectures. This suggests that the best-performing approach for our 14-class temporal relation classification task is the hybrid approach 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.

Among the remaining four architectures, "All Rules with accuracy ≥ 0.8", the version of the rule-based architecture where only the high-accuracy rules are used, performs significantly worse than the others, presumably because the coverage of the ruleset is low. The results of the two feature-based architectures, "Features" and "Features + Rules as Features", are statistically indistinguishable from each other at the $p < 0.01$ level. At the $p < 0.05$ level, however, their results are mixed: "Features + Rules as Features" is better than "Features" according to accuracy, whereas the reverse is true according to macro F-score. Combining these results with those we discussed above concerning the "Rules + Features" and "Rules + Features + Rules as Features" architectures, we can conclude that the features encoding the hand-crafted rules are (mildly) useful only when used in combination with a weak-performing system. Finally, comparing the "Features" architecture and the "All Rules" architecture, we also see mixed results: "Features" is better than "All Rules" according to accuracy, whereas the reverse is true according to macro F-score. These results confirm our earlier hypothesis that the rule-based system is indeed better at identifying instances of minority relation types.

Next, to determine whether the addition of a particular type of features to the feature set is useful, we apply the paired t-test to each pair of adjacent rows in Table 3. We found that adding pairwise features, dependency relations, and most

[8]Unless otherwise stated, all statistical significance tests are paired t-tests, with $p < 0.05$.

	Feature Type	Event-Event Acc	Event-Event F^{ma}	Event-Time Acc	Event-Time F^{ma}
1	Baseline	36.7	15.6	63.3	19.2
2	+ Pairwise	40.4	25.4	64.7	24.2
3	+ Dependencies	42.4	28.4	64.9	25.4
4	+ WordNet	42.6	28.1	64.7	25.3
5	+ Webster	43.0	29.7	64.6	25.3
6	+ PropBank	43.2	28.6	64.3	25.1
7	+ Discourse	46.8	36.3	65.4	26.4

Table 4: Event-event and event-time classification results of our best system (Rules + Features+ Rules as features).

importantly, discourse relations significantly improves both accuracy and macro F-score ($p < 0.05$). Adding the Webster relations improves accuracy at a slightly lower significance level ($p < 0.07$) but does not significantly improve macro F-score. Somewhat counter-intuitively, the WordNet and predicate-argument relations are not useful. We speculate that their failure to improve performance could be attributed to the fact that these relations are extracted by imperfect analyzers. Additional experiments involving the use of gold-standard quality features are needed to precisely determine the reason.

Recall that the results shown in Table 3 were computed over both the order (i.e., event-event) and anchor (i.e., event-time) temporal relations. To gain additional insights into our best-performing system, we show in Table 4 its performance on classifying event-event and event-time relations separately. In comparison to the baseline, both accuracy and macro F-score increase significantly when our system is used in combination with all feature types. In particular, our system yields a relative error reduction of 16–25% for event-event classification and 6–9% for event-time classification over the baseline. The pairwise features, as well as dependency relations and discourse relations, contribute significantly to the classification of both event-event and event-time relations.

Finally, we show in Table 5 the per-class results of the baseline system and our best-performing system. As we can see, our system performs significantly better than the baseline on all relation types, owing to a simultaneous rise in recall and precision.

6 Conclusions

We have investigated a knowledge-rich, hybrid approach to the 14-class temporal relation classification task. Results on the TimeBank corpus show that our approach achieves a relative error reduction of 15–16% over a learning-based baseline that employs a state-of-the-art feature set. Our results suggest that (1) the pairwise features, dependency relations, and discourse relations are useful for temporal relation classification; and (2) hand-crafted rules can better handle the skewed class distribution underlying our dataset via improving minority class prediction. To our knowledge, we are the first to (1) report results for the 14-class temporal relation classification task on TimeBank; (2) successfully employ PDTB-style discourse relations to improve 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. To stimulate research on this task, we make our complete set of hand-crafted rules available to other researchers. We believe that hybrid rule-based and learning-based approaches are promising approaches to language processing tasks that require complex reasoning and hope that they will be given more attention in the community.

	Baseline			Our System		
Relation	R	P	F	R	P	F
Simultaneous	22.5	30.5	25.9	29.5	39.5	33.8
Identity	56.5	51.5	53.9	59.0	57.5	58.2
Before	39.5	38.5	39.0	50.5	50.5	50.5
After	50.5	35.0	41.4	59.5	44.5	50.9
IBefore	0.0	0.0	0.0	32.5	85.5	47.1
IAfter	0.0	0.0	0.0	5.5	50.0	9.9
Includes	54.5	50.5	52.4	61.0	55.5	58.1
Is_Included	71.5	64.5	67.8	74.5	65.0	69.4
During	11.0	31.0	16.2	19.0	34.5	24.5
During_Inv	14.0	20.0	16.5	19.5	40.5	26.3
Begins	4.5	10.0	6.2	37.0	43.5	40.0
Begun_By	6.5	14.5	9.0	35.0	44.0	39.0
Ends	6.5	10.0	7.9	23.5	70.0	35.2
Ended_By	9.0	10.0	9.5	29.0	26.5	27.7

Table 5: Per-class results of the baseline system and our best system (Rules + Features+ Rules as features).

Acknowledgments

We thank the three anonymous reviewers for their detailed and insightful comments on an earlier draft of the paper. This work was supported in part by NSF Grants IIS-1147644 and IIS-1219142. Any opinions, findings, or conclusions expressed in this paper are those of the authors and do not necessarily reflect the views or official policies of NSF.

References

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume: Proceedings of the Demo and Poster Sessions*, pages 173–176.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 33–40.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Eun Young Ha, Alok Baikadi, Carlyle Licata, and James Lester. 2010. NCSU: Modeling temporal relations with markov logic and lexical ontology. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 341–344.

Janet Hitzeman, Marc Moens, and Claire Grover. 1995. Algorithms for analysing the temporal structure of discourse. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics*, pages 253–260.

Judith Klavans and Philip Resnik, editors. 1994. *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*. Association for Computational Linguistics.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2013. A PDTB-styled end-to-end discourse parser. *Natural Language Engineering (to appear)*.

Hector Llorens, Estela Saquete, and Borja Navarro. 2010. TIPSem (English and Spanish): Evaluating CRFs and semantic roles in TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 753–760.

Congmin Min, Munirathnam Srikanth, and Abraham Fowler. 2007. LCC-TE: A hybrid approach to temporal relation identification in news text. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 219–222.

Seyed Abolghasem Mirroshandel and Gholamreza Ghassem-Sani. 2011. Temporal relation extraction using expectation maximization. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 218–225.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*.

Georgiana Puşcaşu. 2007. WVALI: Temporal relation identification by syntactico-semantic analysis. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 484–487.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, David Day, Lisa Ferro, Robert Gaizauskas, Marcia Lazo, Andrea Setzer, and Beth Sundheim. 2003. The TimeBank corpus. In *Corpus Linguistics*, pages 647–656.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the 21st International Conference on Machine Learning*, pages 104–112.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 Task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80.

Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62.