International Journal of Semantic Computing

Finding Temporal Structure in Text: Machine Learning of Syntactic Temporal Relations

Steven Bethard

Department of Computer Science
University of Colorado at Boulder
430 UCB, Boulder, Colorado 80309, USA
bethard@colorado.edu

James H. Martin

Department of Computer Science
University of Colorado at Boulder
430 UCB, Boulder, Colorado 80309, USA
martin@colorado.edu

Sara Klingenstein

Department of Linguistics
University of Colorado at Boulder
295 UCB, Boulder, Colorado 80309, USA
klingesa@colorado.edu

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

This research proposes and evaluates a linguistically motivated approach to extracting temporal structure from text. Pairs of events in a verb-clause construction were considered, where the first event is a verb and the second event is the head of a clausal argument to that verb. All pairs of events in the TimeBank that participated in verb-clause constructions were selected and annotated with the labels BEFORE, OVERLAP and AFTER. The resulting corpus of 895 event-event temporal relations was then used to train a machine learning model. Using a combination of event-level features like tense and aspect with syntax-level features like the paths through the syntactic tree, support vector machine (SVM) models were trained which could identify new temporal relations with 89.2% accuracy. High accuracy models like these are a first step towards automatic extraction of temporal structure from text.

Keywords: Timelines; temporal relations; corpus annotation; machine learning.

1. Introduction

Recent developments in natural language processing have allowed a variety of fine-grained semantic components to be extracted automatically from text. Machine learning systems have shown good performance on a variety of tasks, including the detection of people, organizations and locations [1, 2], as well as the semantic roles these entities play [3, 4]. But there are other important semantic structures that have not yet been addressed. Consider for example a text like:

> Three buildings in the World Trade Center Complex collapsed due to structural failure on the day of the attack. The south tower (2 WTC) fell at approximately 9:59 a.m., after burning for 56 minutes in a fire caused by the impact of United Airlines Flight 175 at 9:03 a.m. The north tower (1 WTC) fell at 10:28 a.m., after burning approximately 102 minutes in a fire caused by the impact of American Airlines Flight 11 at 8:46 a.m. A third building, 7 World Trade Center (7 WTC) collapsed at 5:20 p.m., after being heavily damaged by debris from the Twin Towers when they fell and subsequent fires. [5]

Events like fires burning, planes crashing and buildings collapsing are tied together in this text through a variety of temporal structures. The temporal relations are expressed both explicitly, through words like *after*, and implicitly through inference – the reader knows the South Tower collapsed before the North Tower because the former collapsed at 9:59am and the latter at 10:28am.

Extracting these sorts of temporal structures is crucial for an understanding of the text. Humans perform this kind of extraction in the form of timelines, a kind of summary that matches events with the times at which they occurred. A human might summarize the World Trade Center text above with a timeline like:

8:46am	American Airlines Flight 11 crashes into the North Tower
9:03am	United Airlines Flight 175 crashes into the South Tower
9:59am	The South Tower collapses
9:59am	Debris hits 7 World Trade Center
10:28am	The North Tower collapses
10:28am	Debris hits 7 World Trade Center
5:20pm	7 World Trade Center collapses

But timelines are still natural language text, and machine reasoning requires a much more explicit representation of the temporal structure. Such an explicit representation can be formed by identifying specific words or phrases as the event anchors of the structure, and then drawing explicit temporal relation links between the various events. Figure 1 shows such a machine-oriented representation of the temporal structure of the World Trade Center text. The graph identifies eight important events, covering the various plane impacts, burning fires, and collapsing buildings, and five important times at which these events occurred. These events and times are tied to each other through temporal relations that identify, for example, that the UA 175 impact occurred at 9:03am, and that the impact of AA 11 happened before 1 WTC fell. By breaking the text structure down in this way to its event anchors and the

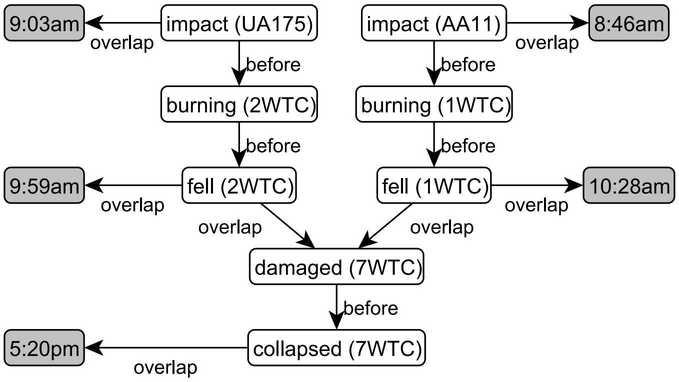

Fig. 1. Temporal structure for the World Trade Center text.

temporal relations between them, this representation provides smaller atomic units which are more easily accessed by machines and which lend themselves better to computational reasoning.

This article shows how computers can be trained to extract these sorts of temporal structures from text using supervised machine learning techniques. It focuses on a particularly common pairing of events, the *verb-clause construction*, and annotates a small corpus of such event pairs with the temporal relations BEFORE, OVERLAP and AFTER. This corpus is then used to train machine learning models which are able to reproduce the human-annotated temporal structure with high levels of accuracy. The remainder of this article is structured as follows. Section 2 discusses some of the findings of previous work and motivates the particular task selected here. Section 3 describes the verb-clause construction, and how a corpus of such event pairs was gathered and annotated for temporal relations. Section 4 explains what features and algorithms were used to train machine learning models on this data, and explains how the models were evaluated. Finally, Section 5 discusses the implications of this work and directions for future research.

2. Prior Work

The natural language processing (NLP) community has recently seen increased interest in extracting important events and temporal structure from text. Researchers have developed a markup language, TimeML, for expressing the structure of events and temporal relations in text [6]. This markup language has allowed human annotators to encode parts of the temporal structure of the 186 document TimeBank corpus to serve as a testbed for automatic techniques [7].

Using the TimeBank corpus, researchers trained models for extracting events and temporal structure. Models trained to find the important events in a document typically combined machine learning techniques with low level features like word stems and parts of speech, and found that important events could be identified with precision and recall in the 70s and 80s [8, 9, 10]. For certain classes of events, like reporting events or verbal events, precision and recall could reach as high as the 90s [8]. On the other hand, identifying the temporal relations between these events proved to be much more difficult. Systems reported some problems with the consistency of the temporal relation annotations in the TimeBank, and found that even in simplified tasks, performance remained in the 50s and 60s [9, 11].

Incorporating such feedback, the creators of the TimeBank organized the 2007 TempEval competition, providing a new set of data annotated for temporal relations [12]. A number of precautions were taken in the TempEval data to facilitate higher inter-annotator agreement. First, TimeML relation types that were hard for annotators to distinguish, like IS-INCLUDED and DURING, were merged into simpler relation types, like OVERLAP. Second, rather than having annotators scan an entire document looking for temporal relations, the annotators were shown one event pair at a time and asked to assign a temporal relation to each. Performing annotation this way meant that annotators would no longer accidentally overlook an important temporal relation.

Of course, most documents contain many events, and annotating all pairs of events with temporal relations would be intractable – trying to assign a temporal relation to two unrelated events many sentences apart would be too difficult for even the best of annotators. Instead, the TempEval competition selected three different types of pairs for annotation:

Task A Events[a] paired with all times in the same sentence. Consider the sentence:

(1) Qantas [$_{\text{EVENT}}$ *plans*] daily flights between Sydney and Bombay to [$_{\text{EVENT}}$ *boost*] business and tourism ties with India, the airline [$_{\text{EVENT}}$ *announced*] [$_{\text{TIME}}$ *Friday*].

In this task, the events *plans*, *boost* and *announced* would each be paired with the time *Friday*.

Task B Events[a] paired with the document creation time. Consider the document:

(2) [$_{\text{TIME}}$ *11/02/89*]
Hadson Corp. [$_{\text{EVENT}}$ *said*] it [$_{\text{EVENT}}$ *expects*] to [$_{\text{EVENT}}$ *report*] a third-quarter net [$_{\text{EVENT}}$ *loss*] of $17 million to $19 million.

In this task, the events *said*, *expects*, *report* and *loss* would each be paired with the document time *11/02/89*.

Task C The matrix verb events of two adjacent sentences. Consider the text:

(3) The action [$_{\text{EVENT}}$ *came*] in response to a petition filed by Timex Inc. Previously, watch imports were [$_{\text{EVENT}}$ *denied*] such duty-free treatment.

In this task, the event *came* would be paired with the event *denied*.

[a] To lower the amount of annotation necessary, only events appearing at least 20 times in the

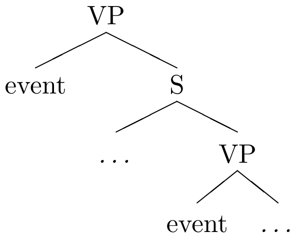

Fig. 2. The verb-clause syntactic construction

A variety of research groups developed systems to compete in these tasks, typically combining machine learning techniques with lexical, syntactic and semantic features [13]. Though performance on Tasks A and C was in the 50s and low 60s, systems trained for Task B reached accuracies as high as the low 80s.

These results suggest that when extracting temporal structure from text, certain types of temporal relations form better starting points than others. Systems performed best in the TempEval Task B, where events were related to the document creation time. This task characterizes the same sorts of relations as the linguistic notion of *tense* (which relates *utterance time* to *event time*). In contrast, on Tasks A and C, whose pairings of events and times were not motivated by linguistic structures, system performance was much poorer.

These results indicate that research on temporal relations has a greater chance of success when guided by linguistic knowledge. Thus, this article explores guiding temporal relation annotation using *syntactic constructions*, or types of patterns on a syntactic tree, that relate one event to another.

3. Corpus Annotation

For this research, temporal relations expressed through the verb-clause construction were examined, as depicted in Figure 2. In the verb-clause construction, the first event is a verb and the second event is the head of a clausal argument to that verb. While this syntactic pattern is fairly specific, it occurs quite frequently – in the TimeBank, almost 50% of adjacent pairs of verbal events participate in exactly a verb-clause construction. For example, consider the sentence:

(4) The top commander of a Cambodian resistance force [$_{\text{EVENT}}$ *said*] Thursday he has [$_{\text{EVENT}}$ *sent*] a team to [$_{\text{EVENT}}$ *recover*] the remains of a British mine removal expert.

This sentence contains two verb-clause constructions: one matching *said* with *sent*, and one matching *sent* with *recover*.

TimeBank were annotated.

A small corpus of these constructions was manually annotated so that automated systems for extracting temporal relations could be trained and evaluated. In order to make a corpus that would be as compatible as possible with previous work, the Wall Street Journal section of the TimeBank was selected for annotation. The 132 newswire documents in this collection had already been manually annotated by other research groups with both gold-standard events from the TimeBank effort [7], and gold-standard syntactic trees from the TreeBank effort [14]. This meant that only a simple set of rules to characterize verb-clause relations in a syntactic tree was necessary to select all of the event-event pairs of interest.

After aligning the TimeBank annotations with the TreeBank annotations, 895 verb-clause pairs of events were extracted from the corpus[b]. These event pairs formed the basic data for annotation. A single annotator first annotated 13 documents (around 50 event pairs) from the middle of the corpus for training. These warm-up annotations were then discarded and the annotator started from the beginning of the corpus, working event-pair by event-pair until all 132 documents (895 event pairs) were annotated. Event pairs were viewed by the annotator one at a time, using a modified version of the TANGO tool which had been used to annotate the TimeBank. Each pair was assigned a label of BEFORE, AFTER or OVERLAP.

The annotation guidelines followed as closely as possible those of TimeBank and TempEval. The labels BEFORE and AFTER were used when it was clear that one event temporally preceded another, and the label OVERLAP was used when the two events occurred over approximately the same time span, or where one event clearly included the other. However, in some cases the TimeBank and TempEval annotation guidelines were not clear enough on how certain words and phrases should be treated. They were therefore augmented with the following additional guidelines:

- Modal or conditional events should be annotated using a possible worlds analysis. So in the example:

  (6) They [$_{\text{EVENT}}$ *expect*] him to [$_{\text{EVENT}}$ *cut*] costs throughout the organization.

  the event pair would be assigned the relation (*expect* BEFORE *cut*) because, in the possible world where costs are *cut*, the *cutting* will have occurred after the *expecting* happening right now.
- Events that perform aspectual-like functions, e.g. describing the manner in which another event is performed, should be annotated as OVERLAP. So in the example:

[b] Originally, 901 verb-clause pairs had been found in the data. Four were removed because an error in the gold-standard syntactic parse had attached a clause to the wrong verb. The other two were removed because they used the connective *unless*, as in:

(5) . . . the best defense lawyers will be [$_{\text{EVENT}}$ *unwilling*] to take criminal cases unless they are [$_{\text{EVENT}}$ *assured*] of being paid.

Since the *unwilling* and *assured* events occur in different, mutually-exclusive possible worlds, attempting to assign a temporal order is extremely difficult (if it even makes sense at all).

	All	Train	Test
Documents	132	94	38
Event pairs	895	581	314
BEFORE relations	368	229	139
OVERLAP relations	344	221	123
AFTER relations	183	131	52

Table 1. Number of documents, event pairs and types of temporal relations in the annotated corpus.

(7) The move may [$_{\text{EVENT}}$ *help*] [$_{\text{EVENT}}$ *prevent*] Martin Ackerman from making a run at the computer-services concern.

the event pair would be assigned the relation (*help* OVERLAP *prevent*) because the *help* event is not really meaningful on its own. It describes how much of the *preventing* the *move* accounts for. (Note that in the TimeBank the event *help* has the I-ACTION class, not the ASPECTUAL class, but it is still considered to be covered by this guideline.)

- Verbs like *allow*, *permit* and *require*, which describe states of affairs that come into being at a single point and continue indefinitely into the future, should be annotated as including only the point at which the state comes into effect. So in the example:

(8) The provision would [$_{\text{EVENT}}$ *require*] BellSouth to [$_{\text{EVENT}}$ *pay*] a price equivalent to what an outside party might have to pay.

the event pair would be assigned the relation (*require* BEFORE *pay*) because the point at which the *requiring* comes into being precedes the point at which the *paying* begins.

These guidelines gave the annotator specific rules for judging some of the more complicated event pairs, and more fully delineated what exactly was meant by the temporal relation labels BEFORE, OVERLAP and AFTER.

All verb-clause event pairs in the corpus were annotated for temporal relations by a single annotator. A small section of this data was also annotated by a second annotator, with the results indicating an inter-annotator agreement of 90% and kappa coefficient of .85. Table 1 shows statistics for this corpus, including the distribution of relation types. For about 40% of the event pairs, the first event was annotated as being BEFORE the second event, for another 40%, the first event OVERLAPped with the second event, and for the remaining 20%, the first event was AFTER the second. The skew against the AFTER relation coincides with the commonly held belief that in narrative text, events are generally mentioned in the order in which they occurred.

The corpus was also split into a training corpus and a testing corpus for the purposes of evaluating machine learning models. Documents wsj_0006-0778 were used for the training set, and documents wsj_0781-1073 were used for the test set. This split placed approximately one third of the temporal relations in the test set and the remaining two thirds in the training set.

4. Machine Learning

The temporal relation identification task presented by this data was framed as a simple three-way classification task. Given a pair of events and an appropriate set of features describing the pair, the classifier was expected to determine whether the first event was BEFORE, OVERLAPing with or AFTER the second event. Since the translation to a classification task was straightforward, much of the effort in developing the temporal relation identification model went into engineering appropriate features. The following section describes the features that were created for this model.

4.1. *Features*

The temporal relation identification task was modeled with two sets of features. The first set gave a linguistic description of isolated events: their tense, aspect, and other basic characteristics of the words. The goal of these features was to be able to handle complex tense-aspect interactions like the differences between Example 9 and Example 10:

(9) Travelers [$_{\text{EVENT}}$ *estimated*] that the California earthquake last month will [$_{\text{EVENT}}$ *cost*] $ 10 million.

(10) Travelers have [$_{\text{EVENT}}$ *estimated*] that the California earthquake last month [$_{\text{EVENT}}$ *cost*] $ 10 million.

In the former, the relation is (*estimated* BEFORE *cost*) because *estimate* appears in the past tense and *cost* appears in the future. In the latter, the relation is (*estimate* AFTER *cost*) because *estimate* appears in the present perfect and *cost* appears in the past. The following features attempted to provide the information necessary to make these kinds of distinctions, and were used once for the first event and once for the second.

word The text of the event itself.

pos The Penn TreeBank gold-standard part-of-speech label for the event, e.g. `NNS` (plural noun) or `VBD` (past tense verb).

stem The morphological stem (or lemma) of the event, e.g. the stem of *approved* is *approve*, determined by a stemming table from the University of Pennsylvania containing around 300,000 words[c]

[c] `http://xbean.cs.ccu.edu.tw/~dan/XTag/morph-1.5/data/`

aux A bag-of-words feature including any auxiliaries and adverbs that modified the event, e.g. in the phrase *hasn't yet been fixed*, the words *has*, *n't*, *yet* and *been* were included.

modal True if any of the auxiliaries were modals, e.g. *will direct* and *couldn't estimate* are MODAL events.

tb-class The TimeBank gold-standard class label for the event, e.g. STATE or REPORTING.

tb-pos The TimeBank gold-standard part-of-speech label for the event, e.g. NOUN or VERB.

tb-tense The TimeBank gold-standard tense label for the event, e.g. PAST or PRESENT.

tb-aspect The TimeBank gold-standard aspect label for the event, e.g. PROGRESSIVE or PERFECTIVE.

tb-polarity The TimeBank gold-standard polarity label for the event, e.g. POS or NEG.

The second set of features for temporal relation identification aimed not at the event words themselves, but at the words that connected one event to the other. Such features are crucial for identifying relations when there are explicit temporal function words like *before* or *after*.

(11) Ratners Group PLC [$_{\text{EVENT}}$ *raised*] its price after another concern [$_{\text{EVENT}}$ *said*] it would be prepared to outbid Ratners's initial offer.

The temporal connective *after* in the example above gives a clear indicator of the expected relation (*raised* AFTER *said*). The following features attempted to characterize these kinds of relations:

compl-word The text of the complementizer for the clause, e.g. *to*, *that* or *because*.

compl-type The type of the complementizer, determined by a simple set of rules such that, for example, *after*, *because* and *since* were grouped under the type AFTER, and *as* and *while* were grouped under the type OVERLAP.

target-path The path of syntactic tree nodes from the clause to its head, e.g. in Figure 3, the path from *running* to *scared* is `S<VP<JJ`.

inter-words A bag-of-words feature including all words between the first event and the second.

func-words A restricted version of the inter-words feature which only included auxiliaries, adverbs, prepositions and other function words, e.g. in the phrase . . . [$_{\text{EVENT}}$ *rose*] *11%, even though claims stemming from Hurricane Hugo* [$_{\text{EVENT}}$ *reduced*] . . . , the function words are *even*, *though* and *from*.

Almost all of the features discussed above relied on either TreeBank or TimeBank annotations. While methods exist for automatically extracting syntactic trees and TimeBank events, in the experiments here, these features were derived from the gold standard annotations for a few reasons. First, most existing systems which can pro-

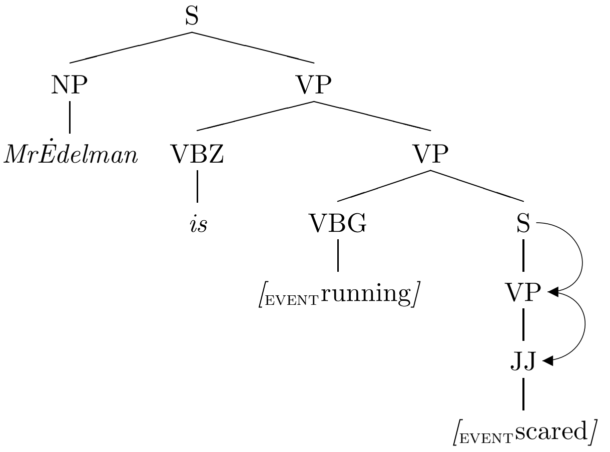

Fig. 3. The target-path from *running* to *scared* is `S<VP<JJ`

duce TimeBank and TreeBank markup were trained on the TreeBank or TimeBank. Thus the performance of these systems on the verb-clause data, which was a subset of the TreeBank and the TimeBank, would have been artificially inflated. Getting results that were not inflated in this way would have required manually annotating verb-clause relations in a second corpus. Second, it was not initially apparent which features would be most important, and since high-performance systems were not available for all parts of the TimeBank markup (e.g. the event semantic classes), it made more sense to begin with the gold standard versions of the features. Knowledge gained of which features were most important for the temporal relations task could then be used to guide future efforts at producing the necessary high-performance systems. The final reason to use the gold standard markup was that it allowed other researchers to more easily reproduce and verify the results reported here. Of course, using gold standard features meant that future work is still needed to determine how having lower quality event and tree annotations effects the model.

4.2. *Model Evaluation*

The features described in the previous section were used to train TinySVM[d] support vector machine classifiers on the training section of the annotated corpus. A one-against-all formulation was used to convert the binary SVM classifiers into multiclass classifiers. To set the model's free parameters, five-fold cross-validations were performed on the training data at a number of different parameter settings, and the best of these were selected as the settings for the final model. These cross-validations set both the cost of misclassification and the degree of the polynomial to 1.0 for the model.

[d] http://chasen.org/ taku/software/TinySVM/

Model	Accuracy
Majority Class	44.3
Tense & Aspect	49.7
SVM	89.2

Table 2. Accuracies of the models on the test data.

	BEFORE		OVERLAP		AFTER	
Model	Precision	Recall	Precision	Recall	Precision	Recall
Majority Class	44.3	100.0	-	0.0	-	0.0
Tense & Aspect	48.3	72.7	54.8	41.5	33.3	7.7
SVM	92.7	91.4	89.0	85.4	81.4	92.3

Table 3. Label precision and recall for the models on the test data.

The performance of the resulting model was compared against two baselines. The Majority Class baseline classified all relations as BEFORE, the class which occurred most frequently in the training data. The Tense & Aspect baseline classified event pairs by looking only at the tense and aspect of the events, and using traditional linguistic analysis. So for example, events in past tense were labeled as being BEFORE events in future tense, and events in the present perfect were labeled as being BEFORE events in the present. Table 2 compares the performance of these two baselines with the SVM model. Both baselines performed poorly on the task, with accuracies less than 50%, while the SVM-based model reached an accuracy of 89.2%, quite close to the inter-annotator agreement of 90%. Table 3 shows a similar trend for performance on the individual labels. The Majority Class model identified only BEFORE relations, with 44.3% precision. The Tense & Aspect model identified mostly BEFORE and OVERLAP relations, with precisions only as high as 54.8%. The SVM model performed dramatically better, with precisions on the various labels between 81.4% and 92.7%, and with recalls between 85.4% and 92.3%.

To determine if these results could be improved by increasing the size of the corpus, SVM models were trained on increasing fractions of the training data. Each such model was evaluated on the testing data to produce the learning curve in Figure 4. The curve leveled off to about 89% model accuracy once about 70% of the training data had been seen, suggesting that the SVM model had already learned most of what it could with the given features by the time it had seen around 400 verb-clause temporal relations.

Since data sparsity was clearly not a problem, an analysis of the features from Section 4.1 was performed to determine which were most useful to the model. For each feature, an SVM model was built using only that feature, and was evaluated through a five-fold cross-validation on the training data[e]. Table 4 shows the ten

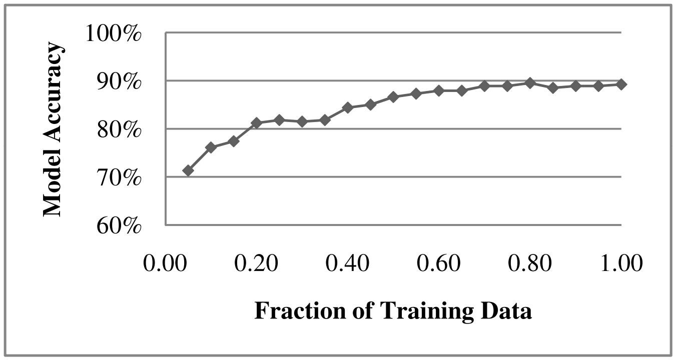

Fig. 4. Learning curve for the SVM model.

Features	Resulting Accuracy
target-path	75.2%
tb-tense (2nd event)	75.0%
pos (2nd event)	71.2%
inter-words	69.7%
aux (2nd event)	69.3%
func-words	65.3%
word (2nd event)	58.8%
compl-word	56.9%
stem (2nd event)	51.7%
stem (1st event)	49.3%
top 10 features above	86.7%
all features	86.9%

Table 4. Cross-validation accuracies for various feature sets. The first ten rows are for single-feature models.

features whose models resulted in the highest cross-validation accuracies. For example, models were able to achieve 75% accuracy by considering only the syntactic path or the tense of the clausal event. Accuracies around 70% were achieved using the part of speech or auxiliaries of the clausal event, or using the words between the two events.

Table 4 also shows the five-fold cross-validation accuracies of the model trained with only the top ten features, as well as the model trained with all the features. The model trained with just the top ten features achieved an accuracy of 86.7%, while using all the features gained only another 0.2%. This suggested that the top ten

[e] The single-feature models were evaluated through cross validations rather than on the test data so as to maintain the validity of the test set for future research.

Feature Missing	Accuracy	ΔA
none (all features)	86.9%	-
inter-words	84.8%	2.1%
tb-class (2$^{\text{nd}}$ event)	85.4%	1.5%
compl-type	85.9%	1.0%
pos (2$^{\text{nd}}$ event)	86.6%	0.3%
target-path	86.6%	0.3%

Table 5. Cross-validation accuracies for models missing various features.

features identified here did most of the job in characterizing the temporal relation identification problem. Interestingly, of the top ten features, four characterized the words connecting the events, five characterized the second (clausal) event, but only one characterized the first (verbal) event. This suggested that for the verb-clause construction, the clausal event plays a more important role than the verbal event in determining the temporal relation. Further research is necessary to investigate this finding.

To get another perspective on the features, a leave-one-out feature analysis was also performed. For each feature, a model was trained using all features except that one. A feature that, when removed, caused a large drop in model performance, was likely to be a feature that the model was depending on more heavily. Table 5 shows the five features which caused the largest performance drops in cross validations on the training data. In general, only a small number of features caused a performance drop when individually removed, and those that did resulted in only very minor decreases in performance. The largest drop in performance was only 2.1% and was caused by removing the feature which identified the words intervening between the two events. These results suggest that the full set of features used here was relatively robust, and that in most cases, when one feature was removed, enough information was still present in the other features to still correctly identify the temporal relations.

As a final analysis of the models, the kinds of errors being made were investigated. First, it was discovered that the models performed substantially worse when one or the other of the events was not a finite verb, like the *plunging* and *development* events in the examples mislabeled by the model below:

(12) The projection [EVENT *sent*] Anheuser shares [EVENT *plunging*] $4.375 in New York Stock Exchange composite trading yesterday.

(13) I [EVENT *think*] it's a pretty positive [EVENT *development*].

In general, on relations where the first event had a TimeBank tense label of NONE, the SVM model accuracy was only 77.3%, and similarly, when the second event had a tense of NONE the SVM model accuracy was only 67.6%. This 10-20% drop from the average model performance suggested that the models were still relying very heavily on the tense of the events to identify the relations between them, and that features that could characterize other aspects of the events were still needed.

The error analysis also showed that the models had more difficulty when the distance between the two events was greater. While on the average in the annotated data there were about 4.7 words between the events in a pair, in the part of the data misclassified by the models, there were about 7.2 words between the events. This often corresponded to the presence of an additional embedded clause or other verbal object intervening between the two events, as in the following examples:

(14) Mr. Fournier [$_{\text{EVENT}}$ *said*] the large institutions *that hold nearly 50% of Navigation Mixte's capital* all strongly [$_{\text{EVENT}}$ *support*] him

(15) GM's North American vehicle production [$_{\text{EVENT}}$ *fell*] *8.4% from a year ago*, which [$_{\text{EVENT}}$ *hurt*] Delco Electronic's earnings

This finding suggested that though the features successfully characterized some of the syntactic relations between the two events, features that focused on only the most important bits of the syntax might still be necessary.

5. Conclusions

This article has presented a new approach to automatically extracting temporal structure from text. Syntactic patterns were used to select pairs of events likely to participate in temporal relations. In particular, the verb-clause construction was considered, which relates almost 50% of adjacent verbal events, for example *said-sent* and *sent-recover* in the following sentence:

(16) The top commander of a Cambodian resistance force [$_{\text{EVENT}}$ *said*] Thursday he has [$_{\text{EVENT}}$ *sent*] a team to [$_{\text{EVENT}}$ *recover*] the remains of a British mine removal expert [$_{\text{EVENT}}$ *kidnapped*] and [$_{\text{EVENT}}$ *killed*] by Khmer Rouge guerrillas almost two years ago.

Using an extension of the TimeBank and TempEval annotation guidelines, all pairs of events in a verb-clause construction in the Wall Street Journal section of the TimeBank corpus were manually annotated with the labels BEFORE, OVERLAP and AFTER. The resulting corpus of 132 documents and 895 event-event temporal relations was then used to train a machine learning model.

By combining event-level features like tense and aspect with syntax-level features like the path between events in a syntactic tree, support vector machine (SVM) models were trained which could identify new temporal relations with 89.2% accuracy, quite close to the 90% level of inter-annotator agreement. Analysis of these models showed that only about 400 event pairs were necessary to reach the maximum performance on this task, and that a small set of features accounted for most of the data. For example, a system considering only the syntactic path between the verbal event and the clausal event performed only about 10% worse than the system considering all the cues. And a model considering only the top ten features,

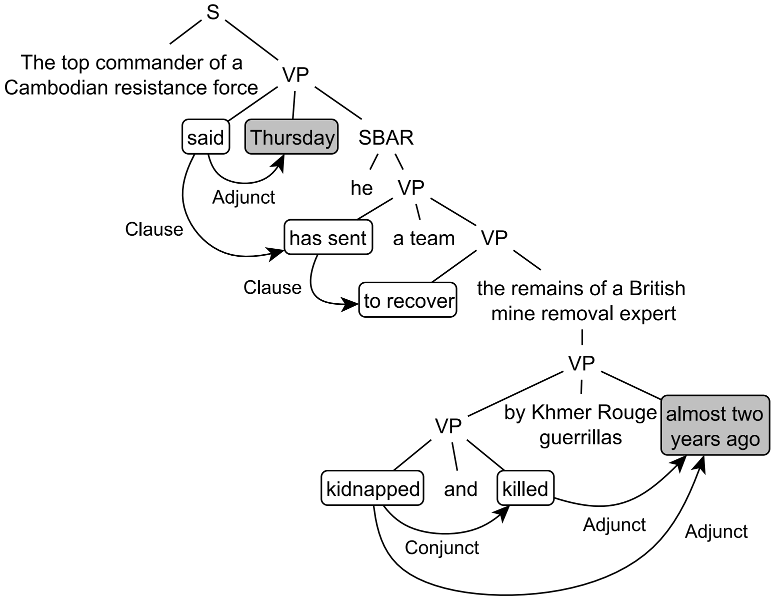

Fig. 5. Temporal structure for Example 16.

which primarily characterized the word level properties of the clausal event and the lexical and syntactic path between the events, performed just as well as a model using all the features.

Two factors help explain the high levels of accuracy achieved by the model. First, the task was carefully constrained to match linguistic intuitions about temporal relations, and the annotation guidelines were explicit about the semantics being annotated. This resulted in high inter-annotator agreement, which is crucial for creating a high-quality corpus for machine learning. Second, the machine learning models were able to take clues from a variety of different lexical and syntactic sources. Simple rule-based models relying on traditional linguistic notions of tense and aspect performed poorly on this task because they were unable to account for all the other factors which played a role in establishing the temporal relation. Thus the combination of a consistently annotated, linguistically motivated task with machine learning methods taking cues from a variety of knowledge sources produced automatic systems capable of accurately identifying temporal relations.

These findings suggest some promising avenues for future research. There are a variety of syntactic constructions that cover other sorts of event-event and event-time pairings. Some notable ones are shown in Figure 5. Conjoined events, like *kidnapped* and *killed*, often indicate either simultaneity or sequence (i.e. either OVERLAP or BEFORE relations). Verbs with temporal adjuncts, like *said* and *Thursday*, almost always express some sort of temporal relation between the event and the time. In fact, PropBank [15] annotates such constructions with the ARGM-TMP

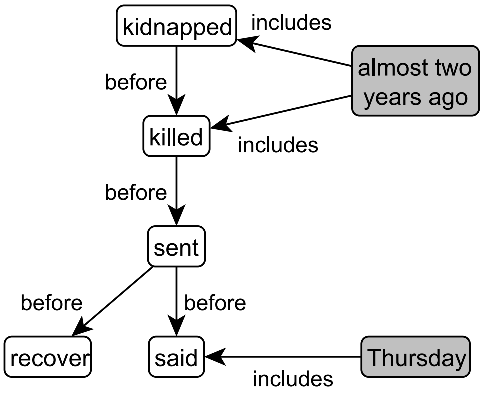

Fig. 6. Temporal structure for Example 16.

role, so a natural extension of the work here would be to annotate each ARGM-TMP role in PropBank with an appropriate BEFORE, OVERLAP or AFTER label.

Together, these various constructions will allow the automatic assembly of a machine-readable temporal structure like that of Figure 6. Joining the pairwise temporal relations together to build such a structure presents some interesting challenges of its own. Since machine learning methods always generate some degree of error, it will be necessary to find some ways to enforce graph consistency. Graph closure algorithms, such as those used by Mani and colleagues [11], will likely play a role here, as will methods for considering the different probabilities assigned to the different relations. Support vector machine models were used in this work and performed well, but do not output actual probabilities, and so investigation into other machine learning methods may be warranted. Thus, much work toward the ultimate goal still remains, but the initial steps taken here will provide a solid foundation for this future research.

Acknowledgments

This research was performed under an appointment of the first author to the Department of Homeland Security (DHS) Scholarship and Fellowship Program, administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and DHS. ORISE is managed by Oak Ridge Associated Universities (ORAU) under DOE contract number DE-AC05-06OR23100. All opinions expressed in this paper are the author's and do not necessarily reflect the policies and views of DHS, DOE, or ORAU/ORISE.

References

[1] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In *Proceedings of CoNLL-2003*, pages 168–171,

2003.

[2] Kadri Hacioglu, Benjamin Douglas, and Ying Chen. Detection of entity mentions occurring in English and Chinese text. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 379–386, 2005.

[3] Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. Support vector learning for semantic argument classification. *Machine Learning*, 60(1):11–39, 2005.

[4] Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. Joint learning improves semantic role labeling. In *ACL 2005*, 2005.

[5] Wikipedia. September 11, 2001 attacks — Wikipedia, the free encyclopedia, 2007. Online; accessed 16-October-2007.

[6] James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. TimeML: Robust specification of event and temporal expressions in text. In *Fifth International Workshop on Computational Semantics (IWCS-5)*, 2003.

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

[8] Steven Bethard and James H. Martin. Identification of event mentions and their semantic class. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

[9] Branimir Boguraev and Rie Kubota Ando. TimeBank-driven TimeML analysis. In Graham Katz, James Pustejovsky, and Frank Schilder, editors, *Annotating, Extracting and Reasoning about Time and Events*, Dagstuhl Seminars. German Research Foundation, 2005.

[10] Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. Evita: A robust event recognizer for QA systems. In *Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*, 2005.

[11] Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. Machine learning of temporal relations. In *International Conference on Computational Linguistics and Association for Computational Linguistics (COLING/ACL)*, 2006.

[12] M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, and J. Pustejovsky. SemEval-2007 task 15: TempEval temporal relation identification. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*, 2007.

[13] Steven Bethard and James H. Martin. CU-TMP: Temporal relation classification using syntactic and semantic features. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*, 2007.

[14] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn TreeBank. *Computational Linguistics*, 19(2):313–330, 1994.

[15] Paul Kingsbury and Martha Palmer. From TreeBank to PropBank. In *Language Resources and Evaluation*, 2002.