Dense Event Ordering with a Multi-Pass Architecture

Nathanael Chambers
United States Naval Academy
nchamber@usna.edu

Taylor Cassidy
Army Research Lab
IBM Research
taylorcassidy64@gmail.com

Bill McDowell
Carnegie Mellon University
forkunited@gmail.com

Steven Bethard
University of Alabama at Birmingham
bethard@cis.uab.edu

Abstract

The past 10 years of event ordering research has focused on learning partial orderings over document events and time expressions. The most popular corpus, the TimeBank, contains a small subset of the possible ordering graph. Many evaluations follow suit by only testing certain pairs of events (e.g., only main verbs of neighboring sentences). This has led most research to focus on specific learners for partial labelings. This paper attempts to nudge the discussion from identifying *some* relations to *all* relations. We present new experiments on strongly connected event graphs that contain $\sim$10 times more relations per document than the TimeBank. We also describe a shift away from the single learner to a sieve-based architecture that naturally blends multiple learners into a precision-ranked cascade of sieves. Each sieve adds labels to the event graph one at a time, and earlier sieves inform later ones through transitive closure. This paper thus describes innovations in both approach and task. We experiment on the densest event graphs to date and show a 14% gain over state-of-the-art.

1 Introduction

Event ordering in the NLP community usually refers to a specific labeling task: given pairs of events and time expressions, choose ordering relations for them. The TimeBank corpus (Pustejovsky et al., 2003) provided a corpus of labeled pairs that motivated machine learning approaches to this task. However, only a small subset of possible pairs are labeled. The annotators identified the most central and obvious relations, and left the rest unlabeled. Partly because of this, there has been little research into complete event graphs. This paper describes our recent exploration into the dramatic changes required of both learning algorithms and experimental setup when faced with the task of a more complete graph labeling.

The most obvious shift required of a complete graph labeling algorithm is that of *relation identification*. Research on the TimeBank and in the TempEval contests has largely focused on *relation classification*. The event pairs are given, and the task is to classify them. We are now forced to first determine which events should be paired up before classification. Our experiments here fully integrate and evaluate both identification and classification.

We are not the first to approach relation identification with classification, but this paper is the first to directly and comprehensively address it. Most recently, TempEval 3 (UzZaman et al., 2013a) proposed a labeling of raw text without prior relation identification, but the challenge ultimately relied on the TimeBank. Systems were only evaluated on its subset of labeled event pairs. This meant that relation identification was largely ignored. The top system optimized only relation classification and intentionally left many pairs unlabeled (Bethard, 2013).

This paper presents CAEVO, a CAscading EVent Ordering architecture. It is a novel sieve-based architecture for temporal event ordering that directly addresses the interplay between identification and classification. We shift away from the idea of monolithic learners, and propose smaller specialized classifiers. Inspiration comes from the recent success in named entity coreference with sieve-based learning (Lee et al., 2013). CAEVO contains a host of classi-

Transactions of the Association for Computational Linguistics, 2 (2014) 273–284. Action Editor: Ellen Riloff.
Submitted 11/2013; Revised 5/2014; Published 10/2014.

fiers that each specialize on different types of edges. The classifiers are ordered by their individual precision, and run in order starting with the most precise. Each sieve informs those below it by passing on its decisions as graph constraints. These more precise constraints are then used by later sieves to assist their less precise decisions.

One of the main advantages of this CAEVO architecture is the natural inclusion of transitive reasoning. After each sieve proposes its labels, the architecture infers transitive links from the new labels and adds them to the graph. This type of expansion has not been adequately evaluated due to our community's sparsely labeled corpora. We are the first to report the surprising result that edges from transitive closure are more precise than the original sieve-provided labels.

Finally, CAEVO enables quick experimentation. Classifiers are pulled in and out with ease, quickly testing new theories. As in D'Souza and Ng (2013), we created rule-based classifiers based on linguistic theory. We naturally integrate these with machine learned classifiers using the sieve architecture. Each classifier focuses on its independent decisions, and the architecture then imposes transitivity constraints. CAEVO contains 12 sieves ordered by precision, and it outperforms the top systems in event ordering.

2 Previous Work

Two lines of research are particularly relevant to our work: research on event ordering annotation, and research on event ordering models. We briefly review both here.

2.1 Event Ordering Annotation

Most prior annotation efforts constructed corpora with only a small subset of temporal relations annotated. In the TimeBank (Pustejovsky et al., 2003), temporal relations were only annotated when the relation was judged to be salient by the annotator. The TempEval competitions (Verhagen et al., 2007; Verhagen et al., 2010; UzZaman et al., 2013b) aimed to improve coverage by annotating relations between all events and times in the same sentence. However, event tokens that were mentioned fewer than 20 times were excluded and only one TempEval task considered relations between events in different sentences.

	Events	Times	Relations	R
TimeBank	7935	1414	6418	0.7
Bramsen 2006	627	–	615	1.0
TempEval 2007	6832	1249	5790	0.7
TempEval 2010	5688	2117	4907	0.6
TempEval 2013	11145	2078	11098	0.8
Kolomiyets 2012	1233	–	1139	0.9
Do 2012[1]	324	232	3132	5.6
This work	1729	289	**12715**	**6.3**

Table 1: Events, times, temporal relations and the ratio of relations to events + times (R) in various corpora.

To avoid such sparse annotations, researchers have explored schemes that encourage annotators to connect all events. Bramsen et al. (2006) annotated timelines as directed acyclic graphs, though they annotated multi-sentence segments of text rather than individual events. Kolomiyets et al. (2012) annotated "temporal dependency structures" (i.e. dependency trees of temporal relations), though they only focused on pairs of events. In Do et al. (2012), "the annotator was not required to annotate all pairs of event mentions, but as many as possible", and then more relations were automatically inferred after the annotation was complete. In contrast, in our work we *required* annotators to label every possible pair of events/times in a given window, and our event graphs are guaranteed to be strongly connected, with every edge verified by the annotators. Table 1 compares the density of relation annotation across various corpora.

A major dilemma from this prior work is that unlabeled event/time pairs are inherently ambiguous. The unlabeled pair holds 3 possibilities:

1. The annotator looked at the pair of events and decided that multiple valid relations could apply.
2. The annotator looked at the pair of events and decided that no temporal relation exists
3. The annotator failed to look at the pair of events, so a single relation may exist.

This ambiguity makes accurate training and evaluation difficult. In the current work, we adopt the VAGUE relation, introduced by TempEval 2007, and force our annotators to indicate pairs for which no clear temporal relation exists. We are thus the first to eliminate the 3rd possibility from our corpus.

[1]Do 2012 reports 6264 relations, but we list half of this because they include relations and their inverses in the count.

2.2 Event Ordering Models

In part because of the sparsity of temporal relations in the available training corpora, most existing models formulate temporal ordering as a pair-wise *classification* task, where each pair of events and/or times is examined and classified as having a temporal relation or not. Early work on the TimeBank took this approach (Boguraev and Ando, 2005), classifying relations between all events and times within 64 tokens of each other. Most of the top-performing systems in the TempEval competitions also took this pair-wise classification approach for both event-time and event-event temporal relations (Bethard and Martin, 2007; Cheng et al., 2007; UzZaman and Allen, 2010; Llorens et al., 2010; Bethard, 2013). These systems have sometimes even explicitly focused on a small subset of temporal relations; for example, the top-ranked ordering system in TempEval 2013 (Bethard, 2013) only classified relations in certain syntactic constructions and with certain relation types.

Systems have tried to take advantage of global information to ensure that the pair-wise classifications satisfy temporal logic transitivity constraints, using frameworks like integer linear programming and Markov logic networks (Bramsen et al., 2006; Chambers and Jurafsky, 2008; Tatu and Srikanth, 2008; Yoshikawa et al., 2009; UzZaman and Allen, 2010). The gains have been small, likely because of the disconnectedness that is common in sparsely annotated corpora (Chambers and Jurafsky, 2008).

An approach that has not been leveraged for event ordering, but that has been successful in the coreference community is the sieve-based architecture. The top performer in CoNLL-2011 shared task was one such system (Lee et al., 2013). The core idea is to begin with the most reliable classifier first, and inform those below it. This idea also appeared in the early IBM MT models (Brown et al., 1993) and in the "islands of reliability" approaches to parsing and speech (Borghesi and Favareto, 1982; Corazza et al., 1991). D'Souza and Ng (2013) recently combined a rule-based model with a machine learned model, but lacked the fine-grained formality of a cascade of sieves. This paper is inspired by the above and is the first to apply it to temporal ordering as an extensible, formal architecture.

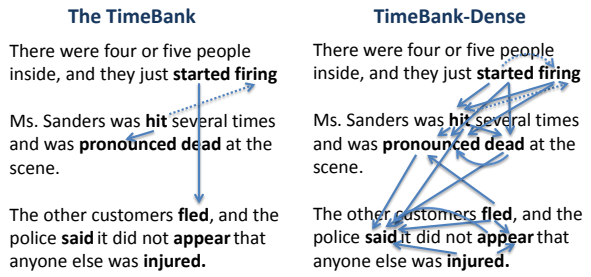

Figure 1: TimeBank document on the left with TimeBank-Dense on the right. Solid arrows indicate BEFORE and dotted INCLUDED IN. Relations with the DCT not shown.

3 TimeBank-Dense: A Dense Ordering

We use a new corpus, called TimeBank-Dense (Cassidy et al., 2014), to motivate and evaluate our architecture. This section highlights its main features.

The TimeBank-Dense corpus was created to address the sparsity in current corpora. It is unique in that the annotators were forced to label all local edges, even in ambiguous cases. The corpus is not a complete graph over events and time expressions, but it approximates completeness by labeling locally complete graphs over neighboring sentences. All pairs of events and time expressions in the same sentence and all pairs of events and time expressions in the immediately following sentence were labeled. It also includes edges between every event and the document creation time (DCT). The document event graphs in TimeBank-Dense are thus strongly connected[2], but they are not complete graphs because edges that cross 2 or more sentences are not explicitly included[3]. See Figure 1 for an illustration of the significant difference between a TimeBank document and a TimeBank-Dense document. The specific edges that were labeled are as follows:

1. Event-Event, Event-Time, and Time-Time pairs in the same sentence
2. Event-Event, Event-Time, and Time-Time pairs between the current and next sentence
3. Event-DCT pairs for every event in the document (DCT is the document creation time)
4. Time-DCT pairs for every time expression in the document

[2]Viewed as an undirected graph: since all events connect the DCT, there exists a path from every event to every other event.

[3]Though many are inferred through transitive closure.

TimeBank-Dense contains 12,715 labeled relations over 36 TimeBank newspaper articles. It uses TimeBank annotated events and time expressions, and then annotates the edges between them. The set of temporal relations are a subset of the 13 original Allen relations (Allen, 1983). The TempEval contests have used both a smaller set of relations (TempEval 1) and all 13 relations (TempEval 3). Published work mirrors this trend, and different groups focus on different aspects of the semantics. The TimeBank-Dense set is a middle ground between coarse and fine-grained distinctions: BEFORE, AFTER, INCLUDES, IS INCLUDED, SIMULTANEOUS, and VAGUE.

The main reason for not using a more fine-grained set is because we annotate pairs that are more ambiguous than those considered in previous efforts. Decisions between relations like BEFORE and IMMEDIATELY BEFORE complicate an already difficult task. The added benefit of a system that makes fine-grained distinctions is also not clear. We lean toward higher annotator agreement by using relations that have a greater separation between their semantics.[4]

Annotators of TimeBank-Dense also had to mutually agree on each labeled edge, otherwise it was labeled VAGUE. Precision between annotators averaged 65.1%, and the kappa ranged from .56 to .64. For comparison, TimeBank annotators averaged 77% precision with a .71 kappa. The breakdown of relation counts is shown in Table 2. **T3** refers to TempEval-3. TempEval-3 used TimeBank documents, but removed a small portion of its events. This paper evaluates on T3 data to maintain a connection to the recent competition, but this resulted in the removal of a portion of the densely annotated edges.

The most significant change in this new corpus is the inclusion of the VAGUE relation. This allows a system to separate relation identification from classification, if desired. The VAGUE relation makes up 46% of the annotated graph. This is the first empirical measurement of how many edges do not carry a clear semantics. This annotation eliminates 1 of the 3 ambiguities (discussed in Section 2) that has plagued ordering research: *the annotator failed to look at the pair of events, so a single relation may exist.*

[4]For instance, a relation like STARTS is a special case of INCLUDES if events are viewed as open intervals, and IMMEDIATELY BEFORE is a special case of BEFORE. We avoid this overlap and only use INCLUDES and BEFORE.

TimeBank-Dense Relation Count

	b	**a**	**i**	**ii**	**s**	**v**	**Total**
All	2590	2104	836	1060	215	5910	12715
T3 train	1444	1148	473	629	120	3013	6827
T3 dev	242	218	37	73	20	359	949
T3 test	589	428	116	159	39	900	2231

Table 2: The number of each relation type in the TimeBank-Dense corpus. T3 is the TempEval3 version used in this paper's experiments.

# of Mutual Vague	Partial Vague	No Vague
1657 (28%)	3234 (55%)	1019 (17%)

Table 3: VAGUE relation origins. Partial vague: one annotator does not choose vague. No vague: neither annotator chooses vague, but they disagree on the specific relation.

This is a significant improvement over previous ordering annotations. Several questions can now be answered to benefit learning algorithms. Which set of edges in the graph are unambiguous? Can I know for certain that I have all BEFORE relations in this document? These questions are critical to supervised learning, and we see the removal of this ambiguity as an important step toward better classification.

Of course, an ambiguity still remains: does VAGUE mean that no relation exists, or that multiple relations are possible? This is not unique to our corpus, but is present in all current ordering corpora. We erred on the side of caution because it is not clear how this can be resolved. Though we don't solve this problem here, we can more deeply analyze how VAGUE relations come about during annotation. Table 3 shows the 3 annotation scenarios that result in a VAGUE relation, and how often each occurred: (1) mutual vague: annotators agreed on VAGUE, (2) partial vague: one annotator chose VAGUE but the other chose a non-vague relation, and (3) no vague: annotators chose different non-vague relations. These disagreements will be released with the corpus, enabling future research into distinguishing these fine-grained semantics.

TimeBank-Dense is split into a 22 document training set, 5 document dev set, and a 9 document test set [5]. We follow this split in all experiments.

[5]Available at http://www.usna.edu/Users/cs/nchamber/caevo/

4 Sieve-Based Temporal Ordering

The sieve architecture applies a sequence of temporal relation classifiers, one at a time, to label the edges of a graph of events and time expressions. The individual classifiers are called sieves, and each sieve passes its temporal relation decisions onto the next sieve. The input to each sieve is the partially labeled graph from previous sieves. These previous decisions can help inform a sieve's own decisions. The sieves are ordered by precision, running the most precise models first. Events and time expressions are assumed to be annotated with their standard attributes (such as verb tense). This work uses gold event attributes for consistent analysis of ordering decisions.

One of the benefits of this architecture is the seamless enforcement of transitivity constraints. Independent pairwise classifiers often produce inconsistent labelings. The architecture avoids inconsistent graphs by inferring all transitive relations from each sieve's output before the graph is passed on to the next one. Specifically, the n^{th} sieve cannot add an inconsistent edge that is inferrable from the previous $n-1$ sieves. Transitive inference is run after each stage, so it is not possible to relabel an already inferred edge. In a similar fashion, the architecture enforces the *precision preference rule*: the n^{th} sieve may not label edge e if one of the previous $n-1$ sieves has already labeled it. Transitive inferences are made with all possible relations (e.g., A before B and B includes C infers A before C). The only relation that does not infer new edges is VAGUE.

This architecture facilitates a seamless integration of a wide variety of classifiers. We experimented with both rule-based and machine learned classifiers. The following subsections describe the final set of 12 temporal classifiers in CAEVO. We begin with the deterministic rule-based sieves.

4.1 Deterministic Sieves

The most precise sieves come from linguistic insight into syntactic and semantic patterns. Many pairs of events share common syntactic attributes that clearly indicate a temporal relation. Machine learning is not needed for these known linguistic phenomena, and a trained classifier can incorrectly label these obvious pairs if other features incorrectly override the decision. We therefore rely on deterministic rule-based approaches to capture these phenomena.

4.1.1 Verb/Timex Adjacency

Many prepositions in English carry either explicit (e.g., before, during) or implicit temporal semantics (e.g., in, over). Several are so reliable that a simple series of rules can label event-timex edges with very high precision. This Verb/Timex Adjacency sieve thus applies to event words and time expressions that are connected by a direct path in the syntactic parse tree, as well as all event/time pairs such that the event is at most 2 tokens before the time expression.

Time expressions under a preposition are handled with specific rules. For instance, *in, on, over, during*, and *within* all resolve to IS INCLUDED. The prepositions *for, at*, and *throughout* resolve to SIMULTANEOUS. In the absence of a preposition, a timex might simply be a modifier of the verb. The following is one such example.

> Police confirmed Friday that the body found along a highway...

The *confirmed* event IS INCLUDED in *Friday*. The edge between an event and a direct time modifier all default to IS INCLUDED. The precision of this sieve on the development set is 92%, the top performer.

4.1.2 Timex-Timex Relations

One of the most precise sieves is a small set of deterministic rules that address relations between time expressions (including the DCT). This sieve uses the normalized time values for time expressions to compare their start/end times. Given two time spans with defined starting and ending time points, the temporal relation is unambiguous.

The more difficult cases involve time expressions without stated start/end points. Examples of these include references to the abstract future or past, such as in phrases like 'last year' and 'next quarter'. This sieve chooses BEFORE if the abstract references are not the same, but in clear opposition to each other. For instance, a past reference and a future reference are labeled BEFORE. If the abstract references are the same, then VAGUE is typically applied. We defer to the publicly released code for further details.

4.1.3 Reporting Event with DCT

Newspaper articles contain an abundance of reporting events (e.g., said, reply, tell). We crafted a

targeted sieve to handle this genre-specific phenomena. Since the document creation time is often the day of the reporting events, edges between reporting events and the DCT are labeled as IS INCLUDED. This is essentially a high baseline for event-DCT edges. One of the advantages of the sieve-based architecture is that this sieve can be placed later in the cascade of sieves. More sophisticated reasoners run before this one, making more informed decisions when appropriate, and this sieve cleans up the remaining reporting-DCT edges that are still unlabeled.

4.1.4 Reporting Event with Dominated Event

We also created rules for edges between a reporting event and another event that is syntactically dominated by the reporting event. Statistical classifiers can pick up patterns for reporting verbs, but the labels are consistent enough that a few deterministic rules can handle many of the cases. The following rules are solely based on the tense and grammatical aspect of each event.

The following is a sampling of these rules, where *gov* is the governing event and *dep* the dependent.

1. If gov is present, and dep is past
   then (gov AFTER dep)
2. If gov is present, and dep is present perfect
   then (gov AFTER dep)
3. If gov is present, and dep is future
   then (gov BEFORE dep)
4. If gov is past, and dep is past perfective
   then (gov AFTER dep)
5. If gov is past, and dep is past progressive
   then (gov IS INCLUDED dep)

4.1.5 General Event with Dominated Event

We generalized the above reporting-event sieve to apply to all types of events. This 'backoff' sieve handles the non-reporting event pairs where one event immediately dominates another. It operates similar to the preceding sieve and uses event features like tense and aspect to make its decisions. However, unlike the preceding, the decisions are split into different rule sets based on the specific typed dependency relation that connects the two events. We have separate rules nsubj, dobj, xcomp, ccomp, advcl, and conj.

4.1.6 Reichenbach Rules

Reichenbach (1947) defined the role played by the various tenses of English verbs in conveying temporal information in a discourse. Each tense is distinguished in terms of the relative ordering of the following time points:

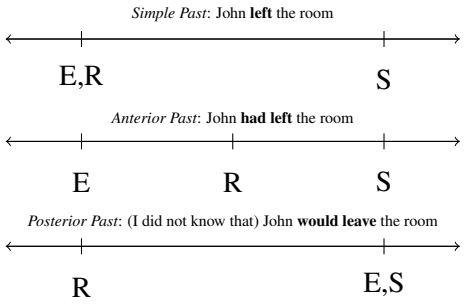

Figure 2: Timelines in which the ordering $S < R$ is fixed because the verb *to leave* is in the past tense.

- ***Point of speech*** **(**S**)** The time at which an act of speech is performed.
- ***Point of event*** **(**E**)** The time at which the the event referred to by the verb occurs.
- ***Point of reference*** **(**R**)** A single time with respect to which S is ordered by one dimension of tense, and to which E is ordered by another.

Reichenbach's account maps the set of possible orderings of S, E, and R, where each pair of elements can be ordered with $<$ (before), $=$ (simultaneous), or $>$ (after), onto a set of tense names given by: $\{anterior, simple, posterior\} \times \{past, present, future\}$. The relative ordering of E and R is indicated by the first dimension, while that of S and R is given by the second. Figure 2 depicts examples of the ordering $R < S$.

Similar to Derczynski and Gaizauskas (2013) we map a subset of Reichenbach's tenses onto pairs of TimeML tense and aspect attribute values, which are given by $\{simple, perfect, progressive\} \times \{past, present, future\}$, where the first dimension is *grammatical aspect* and the second is *tense*. We refer to an event's tense/aspect combination as its *tense-aspect profile*. Consider two events e_1 and e_2 associated with $S_1/E_1/R_1$ and $S_2/E_2/R_2$. Intuitively, given $R_1 = R_2$ and the time point orderings for each event (derived from its tense-aspect profile) we can enumerate the possible interval relations that might hold between E_1 and E_2, which we can interpret as a disjunction of the possible orderings of e_1 and e_2 (Derczynski and Gaizauskas, 2013). We identify a subset of tense/aspect profile pairs that

unambiguously yield an interval relation from our 6 relations, and implement them as rules in the *Reichenbach Sieve*. Three such rules are given here:

1. If $e1$ is *past/simple* and $e2$ is *past/perfect* then ($e1$ AFTER $e2$)
2. If $e1$ is *future/simple* and $e2$ is *present/perfect* then ($e1$ AFTER $e2$)
3. If $e1$ is *past/simple* and $e2$ is *future/simple* then ($e1$ BEFORE $e2$)

The Reichenbach sieve achieved a precision of 61% on the entire dataset, and 91% when links labeled VAGUE were excluded, a notably larger increase over other sieves. The lower 61% is primarily due to the abundance of non-canonical usages of various tense/aspect profiles. For example, the present tense may convey habitual action, an event conveyed with the future tense in a quotation may have already occurred, the present perfect may be used to indicate that an event occurred at least once in the past, etc.

4.1.7 WordNet Rules

Determining whether two words are precisely co-referent is notoriously difficult (Hovy et al., 2013). Synonymous words, or even those sharing the same surface form may be only partially co-referent and thus likely labeled VAGUE by annotators. For example, news articles state what someone *said* multiple times without revealing temporal ordering between the events. Event-event edges whose events share the same lemma, or are each a member of the same synset are thus labeled VAGUE. Time-time edges with the same lemma/synset are labeled SIMULTANEOUS.

4.1.8 All Vague

The majority class baseline for this task is to label all edges as VAGUE. This sieve is added to the end of the gauntlet, labeling any remaining unlabeled edges.

4.2 Machine Learned Sieves

Current state-of-the-art models for temporal ordering are machine learned classifiers. The top systems in the latest TempEval-3 contest used supervised classifiers for the different types of edges in the event graph (Bethard, 2013; Chambers, 2013). In the spirit of the sieve architecture, rather than training one large classifier for all types of edges, we create targeted classifiers for each type of edge. The resulting models are again ranked by precision and mutually inform each other through the architecture's transitive constraints. All models use the MaxEnt classifier from the Stanford CoreNLP with its default settings. Below we describe the features for each sieve.

Event-Time Features

Token, lemma, POS tag of event
Tense, aspect, class of event
Bigram of event and time expression words
Token path from event to time
Syntactic parse tree path between the event and time
Typed dependency edge path between the events
Boolean: syntactically dominates or is-dominated
Boolean: time expression concludes sentence?
Boolean: time expression day of week?
Full time expression phrase

Figure 3: Features for event-time edges.

4.2.1 Event-Time Edges

We created two different sieves for labeling edges between an event and a time expression. Similar to the event-event distinction, one classifier applies to intra-sentence edges and the other to inter-sentence. The features for these classifiers are shown in Figure 3. The same features from the above event-event classifier are used where applicable. These include the parse paths, syntactic dominance, and event features with POS tags and token context. Features specific to event-time edges include the event word's tense, aspect, and class (included independently from the time expression). Time features included a boolean feature if it was a day of the week, the head word of the expression, and the entire expression if it is more than one word. A boolean feature indicating if the time expression ended the sentence is also included.

4.2.2 Event-Event Edges

We created three different sieves for event-event edges. They use the same features, but are trained and applied to different subsets of the graph edges.

1. intra-sentence: Label edges between all event pairs that occur in the same sentence.
2. intra-sentence dominance: Label edges between all event pairs that occur in the same sentence *and* one event dominates the other in the syntactic parse tree.
3. inter-sentence: Label edges between all event pairs that occur in neighboring sentences.

Event-Event Features

Text order normalized (put the first event first)
Boolean: syntactically dominates or is-dominated
Boolean: is there an event in between these two?
Syntactic parse tree path between the events
Typed dependency edge path between the events
Token n-grams, POS tag n-grams
Pairs of tense, aspect, event class
Prepositions attached to each event, if any

Figure 4: Features for intra-sentence event-event edges.

The features for these classifiers are summarized in Figure 4. The POS tag features for each event include its previous two tags, the event word's tag, and the POS bigram ending on the event word. A POS bigram is also created from the POS tag of each event. The token word, its lemma, its WordNet synset, and the token bigram from each event word are also used. Various syntactic features are used, including preposition words that dominate either event, boolean dominance features if one event is above the other in the parse tree, the parse path between the two events (append the non-terminals together), and the typed dependency path between the two. Finally, we use event attributes of tense, aspect, and class to create event pairs of their values (e.g., if both events are past tense, then the feature's value is 'past past').

The final CAEVO architecture includes both intra-sentence learners, but not inter-sentence. The latter's precision was not high enough to warrant its inclusion. As mirrored in recent TempEval competition results, inter-sentence event relations continue to be an area of future research.

4.2.3 Event-DCT Edges

Temporal relations between events and the document creation time (DCT) are also separately classified. The features for this machine-learned sieve focus solely on the event word, including its POS tag, the two previous POS tags, event lemma, WordNet synset, token unigrams from a window of size 2, and event attributes: tense, aspect, and class. Since this sieve relies entirely on the event's features with no DCT specific features, it can be viewed as a single-event classifier. We also experimented with a number of rule-based sieves for Event-DCT classification, but they were all outperformed by the machine-learning sieve on the training and development sets.

4.3 The Complete Sieve Gauntlet

We now present the sieves used by CAEVO. Although the final system contains 12 sieves, we experimented with over 25 different classifiers. The final set of 12 are sorted by precision:

$$precision(sieve) = \frac{\#\text{correct}}{\#\text{proposed by } sieve} \quad (1)$$

Transitive closure is not included when calculating precision. Sieves are ranked according to precision, placed in order from highest to lowest. Sieves with a precision lower than the All Vague baseline were removed. Table 4 shows the final sieve order.

5 Experiments and Results

We now present the first experiments on the TimeBank-Dense corpus, using the set of events used for TempEval-3. These experiments highlight two separate contributions. The first is a new approach to temporal ordering, the CAEVO architecture with a cascade of classifiers. The second is an evaluation of an ordering system on the densest event graphs to date. We provide per-sieve results and compare against two state-of-the-art systems.

All sieves were developed on and motivated by the TimeBank-Dense training data. Precision was initially computed on the development set, and final sieve ordering is based on individual performance. Final performance is reported on the test set.

We compare against the *All Vague* baseline that labels all edges as VAGUE. We also compare against the winner of TempEval-3, ClearTK (Bethard, 2013). We include ClearTK in four configurations: (1) the base TempEval-3 system, (2) with CAEVO transitivity, (3) with transitivity and altered to output a VAGUE relation where it normally would have skipped the decision, and (4) retrained models on TimeBank-Dense with transitivity and all VAGUE. We also include the second place TempEval-3 system, NavyTime (Chambers, 2013), retrained and run as (3) and (4). Results are shown in Table 5. The optimized TempEval-3 systems ultimately achieve ∼44 F1. The transitivity architecture automatically boosted ClearTK from .147 to .264 with no additional changes. CAEVO outperforms at 50.7 F1, improving over the retrained TempEval-3 systems by 14% relative F1.

Ordered Sieves

		Dev		Test	
Sieve	**Brief Description**	**P**	**R**	**P**	**R**
Verb/Time Adjacent	Rules for edges between one verbal event and one Timex	.92	.01	.74	.01
TimeTime	Rules for edges between two Timex	.88	.02	.90	.02
Reporting Governor	Rules for when a reporting event dominates an event	.69	.01	.91	.01
Reichenbach	Rules following Reichenbach theory for edges between two events	.63	.02	.67	.03
General Governor	Rules for edges where one event dominates another via typed dependencies	.63	.04	.51	.02
WordNet	Rules for edges between two events/times based on WordNet synonym lookup	.52	.02	.57	.01
Reporting DCT	Rules for reporting events and the DCT	1.0	.00	1.0	.00
ML-E-T SameSent	Trained classifier for event-time edges	.50	.03	.42	.03
ML-E-E SameSent	Trained classifier for all intra-sentence event-event edges	.45	.11	.44	.10
ML-E-E Dominate	Trained classifier for edges where one event syntactically dominates another	.41	.05	.44	.05
ML-E-DCT	Trained classifier for all event-DCT edges	.40	.06	.53	.07
AllVague	Labels all edges as VAGUE	.38	.38	.40	.40

Table 4: The final order of sieves in CAEVO. Precision was computed on the development set. ReportingDCT only matched one pair in the dev set, so we moved it below the other more common rule-based sieves.

System Comparison

System	**P**	**R**	**F1**
ClearTK	.397	.091	.147
ClearTK + trans	.390	.199	.264
ClearTK VAGUE + trans	.431	.431	.431
ClearTK Dense VAGUE + trans	.460	.426	.442
NavyT Dense VAGUE	.485	.415	.447
NavyT Dense VAGUE + trans	.483	.427	.453
Baseline: All Vague	.405	.405	.405
CAEVO	**.508**	**.506**	**.507**

Table 5: Comparison with top systems. The "Dense" systems were retrained on TimeBank-Dense. The "Opt" systems were optimized to label unknown edges as *VAGUE*.

CAEVO Component Ablation

	P	**R**	**F1**
CAEVO with only ML Sieves	.458	.202	.280
+ Rule-Based Sieves	.486	.240	.321
+ Transitivity	.505	.328	.398
+ All Vague Sieve	.507	.507	.507
Direct Sieve Predictions	.498	.398	.442
Predictions from Transitivity	.544	.109	.182

Table 6: System precision on the test set with and without transitivity, and precision of transitive labels.

The bottom of Table 6 illustrates the contribution of transitive closure in the labeling decisions. Prior research has hypothesized that transitive closure should help individual decisions (Chambers and Jurafsky, 2008; Yoshikawa et al., 2009), but this has not been adequately tested due to the sparsely labeled graphs available to the community. We computed the precision of edges directly labeled by sieves, and the precision of edges inferred from transitive closure. Transitive inferences achieve *better* performance than directly classified edges (54.5% to 49.8% precision).

Table 7 breaks apart precision by edge type. Since CAEVO labels all edges, precision and recall are the same. Edges between an event and the DCT show 55% precision, while event-event and event-time edges achieve the lowest performance at 49%. Linking time expressions to each other is helpful. While far less numerous, time-time edges are the easiest to classify (71% precision). Upon removing the Time-Time sieve, overall results drop almost 5%, despite the time-time edges only making up a smaller 2.6% of the data. Combined with transitive closure, they positively influence the other event decisions.

Finally, Table 8 shows CAEVO's per-relation results. We also reversed the order of sieves (but with all vague still last) to see what effect it has. Performance on the dev set dropped from .48 F1 to .46.

6 Discussion

These are the first experiments on event ordering where the document event graphs were annotated so as to guarantee high density of relations and strong

Edge Type Performance

Edge Type	Total	P/R/F1
Event-Event Edges	1427	.494
Event-Time Edges	423	.494
Event-DCT	311	.553
Time-Time Edges	59	.712

Table 7: The number of edges for each edge type, and the system's overall precision on each. Since CAEVO labels all possible edges, precision and recall are the same.

Per-Relation Performance

Relation	P	R	F1
BEFORE	.52	.45	.49
AFTER	.55	.38	.45
INCLUDES	.44	.21	.28
IS INCLUDED	.57	.43	.49
SIMULTANEOUS	.71	.31	.43
VAGUE	.48	.66	.56

Table 8: Performance on individual relation types.

connectivity. Most related to our experiments is the dense annotation work of Do et al. (2012), but as they did not force annotators to label every pair, unannotated pairs in their corpus are ambiguous between a VAGUE relation and a relation that was simply missed by the annotators. This ambiguity causes problems for evaluation that are not present in a corpus where annotators were required to annotate all pairs.

Constructing event ordering systems that will be evaluated on dense, strongly connected document event graphs is fundamentally different than optimizing a system to label a subset of edges. This can be seen in the performance of the ClearTK-TimeML system, which achieved the top performance (36.26) on the sparse relation task of TempEval 2013, but performed dramatically worse (15.8) on our dense relation task. This is not to fault ClearTK, but to illustrate the difference of evaluating only a subset of an event graph's edges rather than the entire graph.

We also presented the first sieve-based architecture for event/time ordering. The CAEVO architecture allowed us to focus on specific classifiers for local decisions, and leave constraint enforcement to the architecture itself. There are several intangible benefits to this setup that don't appear in the results: (1) seamless integration of new/existing classifiers, (2) extremely quick ablation experiments by adding/removing sieves, and (3) facilitation of the development of specific classifiers that are agnostic to global constraints.

The final results are encouraging. Our multi-pass system achieved an F1 of 50.7, a full 10 F1 points over the All-Vague baseline, and a 14% relative increase over the top two systems in TempEval 3. This increase is based not just on the top systems out of the box, but after retraining them with TimeBank-Dense. The largest factor contributing to this gain is the enforcement of global constraints. Transitive inference alone boosted performance by about 10%.

Finally, we note that a lot of effort was put into linguistically motivated rule-based classifiers. The overall gain from these high precision sieves was 2 F1 points. The Reichenbach sieve's application of a more traditional linguistic theory of time helped increase performance. Error analysis reveals there are inherent difficulties in relying solely on tense and aspect for ordering decisions, but we hope to continue experiments into this and other theories.

CAEVO and its code is now publicly available[6]. In addition to temporal classification (the focus of this paper), it also automatically extracts events and time expressions. CAEVO is a complete system that produces dense event/time graphs from raw text. We hope it provides a new platform for quick experimentation and development. The TimeBank-Dense corpus is also publicly available, as well as the new annotation tool that forces annotators to label all edges. It is a unique tool that helps with transitive inference as annotators work. We hope it encourages further dense data creation for temporal research.

Acknowledgments

This work was supported, in part, by the Johns Hopkins Human Language Technology Center of Excellence. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors. It was also supported by Grant Number R01LM010090 from the National Library Of Medicine. Finally, thanks to Benjamin Van Durme for assistance and insight.

[6]http://www.usna.edu/Users/cs/nchamber/caevo

References

James F Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

Steven Bethard and James H. Martin. 2007. CU-TMP: Temporal relation classification using syntactic and semantic features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 129–132, Prague, Czech Republic, June. Association for Computational Linguistics.

Steven Bethard. 2013. ClearTK-TimeML: A minimalist approach to tempeval 2013. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 10–14, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Branimir Boguraev and Rie Kubota Ando. 2005. TimeBank-driven TimeML analysis. In *Annotating, Extracting and Reasoning about Time and Events*. Springer.

Luigi Borghesi and Chiara Favareto. 1982. Flexible parsing of discretely uttered sentences. In *Proceedings of the 9th conference on Computational linguistics-Volume 1*, pages 37–42. Academia Praha.

Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006. Inducing temporal graphs. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 189–198. Association for Computational Linguistics.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *The 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, MD, USA, June. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 698–706. Association for Computational Linguistics.

Nathanael Chambers. 2013. Navytime: Event and time ordering from raw text. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Yuchang Cheng, Masayuki Asahara, and Yuji Matsumoto. 2007. NAIST.Japan: Temporal relation identification using dependency parsed tree. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 245–248, Prague, Czech Republic, June. Association for Computational Linguistics.

A Corazza, R De Mori, R Gretter, and G Satta. 1991. Stochastic context-free grammars for island-driven probabilistic parsing. In *Proceedings of Second International Workshop on Parsing Technologies (IWPT91)*, pages 210–217.

Leon Derczynski and Robert Gaizauskas. 2013. Empirical validation of reichenbach's tense framework. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, pages 71–82.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687, Jeju Island, Korea, July. Association for Computational Linguistics.

Jennifer DSouza and Vincent Ng. 2013. Classifying temporal relations with rich linguistic knowledge. In *Proceedings of NAACL-HLT*, pages 918–927.

Eduard Hovy, Teruko Mitamura, Felisa Verdejo, Jun Araki, and Andrew Philpot. 2013. Events are not simple: Identity, non-identity, and quasi-identity. *NAACL HLT 2013*.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2012. Extracting narrative timelines as temporal dependency structures. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 88–97, Jeju Island, Korea, July. Association for Computational Linguistics.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*.

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, Uppsala, Sweden, July. Association for Computational Linguistics.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40.

Hans Reichenbach. 1947. *Elements of Symbolic Logic*. Macmillan.

Marta Tatu and Munirathnam Srikanth. 2008. Experiments with reasoning for temporal relations between events. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 857–864. Association for Computational Linguistics.

Naushad UzZaman and James Allen. 2010. TRIPS and TRIOS system for TempEval-2: Extracting temporal information from text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 276–283, Uppsala, Sweden, July. Association for Computational Linguistics.

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

Naushad UzZaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen, and James Pustejovsky. 2013b. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. Association for Computational Linguistics.

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 4th International Workshop on Semantic Evaluations*, pages 75–80. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Katsumasa Yoshikawa, Sebastian Riedel, Masayuki Asahara, and Yuji Matsumoto. 2009. Jointly identifying temporal relations with Markov Logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 405–413. Association for Computational Linguistics.