Temporal Information Extraction

Xiao Ling and **Daniel S. Weld**

Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350, U.S.A.
{xiaoling, weld}@cs.washington.edu

Abstract

Research on information extraction (IE) seeks to distill relational tuples from natural language text, such as the contents of the WWW. Most IE work has focussed on identifying static facts, encoding them as binary relations. This is unfortunate, because the vast majority of facts are fluents, only holding true during an interval of time. It is less helpful to extract `PresidentOf(Bill-Clinton, USA)` without the temporal scope `1/20/93 - 1/20/01`. This paper presents TIE, a novel, information-extraction system, which distills facts from text while inducing as much temporal information as possible. In addition to recognizing temporal relations between times and events, TIE performs global inference, enforcing transitivity to bound the start and ending times for each event. We introduce the notion of *temporal entropy* as a way to evaluate the performance of temporal IE systems and present experiments showing that TIE outperforms three alternative approaches.

Introduction

Information extraction (IE), the problem of extracting relational data from unstructured text, is gaining increased attention by researchers who seek to distill knowledge for the vast corpus of natural-language content on the WWW. Bootstrapped pattern learners (Brin 1998; Riloff and Jones 1999; Agichtein and Gravano 2000), supervised learning (Craven et al. 2000), human-engineered rules (Suchanek, Kasneci, and Weikum 2007), self-supervised probabilistic sequential models (Banko et al. 2007; Wu and Weld 2007), and numerous other approaches (Etzioni et al. 2005) have been used to extract facts from text. On reflection, however, almost all research on information extraction has focused on the acquistion of static (time invariant) facts. In the cases where fluents[1] *were* extracted, *e.g.*, `employed-by(mark-craven, carnegie-mellon-univ)`, temporal arguments were neglected. The sparsity of research on temporal extraction is surprising, since so many statements are temporally qualified. In particular, sources such as newswire text or Wikipedia are predominantly temporal.

---

[1]Fluents are dynamic relations whose truth is a function of time (Russell and Norvig 2003).

We don't mean to suggest that scientists have neglected the challenge of understanding temporal expressions in natural language; indeed, the literature is vast. However, most research on the topic has focused on subproblems rather than the complete task of temporal IE. While we discuss related work more fully in the next section, we now highlight the 2007 TempEval challenge (Verhagen et al. 2007), which asked researchers to identify event-time and event-event relations using a restricted set of Allen-style (Allen 1983) interval comparitors. While TempEval has greatly spurred research on temporal NLP, progress on its challenge does not necessarily lead directly to gains on the broader task of temporal IE due to several of TempEval's simplifying assumptions. First, the TempEval corpus provides researchers with gold-standard annotations of all relevant events, temporal expressions and document creation times. Secondly, the three temporal relations used in TempEval (BEFORE, AFTER and OVERLAP) are insufficient to precisely bound the start and ending points of the events (i.e. OVERLAP is ambiguous) — as would be required, for example, to create a comprehensive timeline of events.

Temporal Information Extraction   In contrast, we focus on the following problem. Given a corpus of natural language text, T, output a set of temporal elements E and temporal constraints C subject to the following conditions. Every element in E should denote an event or a time. We use the notion of temporal element to unify the notion of times and events — indeed, one may view a temporal reference (e.g., "1990") in the same way as an event, denoting an interval of time.[2] For every element, $e \in E$, we refer to its beginning and ending time points, $^\triangleleft e$ and $e^\triangleright$, respectively.

The constraints in C are linear inequalities between time points.[3] For example, the sentence "Steve Jobs revealed the iPhone in 2007." might produce the following constraints:

$$^\triangleleft \text{Year-2007} \le {}^\triangleleft \text{Reveal(Jobs, iPhone)}$$
$$\text{Reveal(Jobs, iPhone)}^\triangleright \le \text{Year-2007}^\triangleright$$

At present, we restrict our attention to reasoning within one sentence at a time. Our objective is to output a maximal set

---

[2]As has been previously observed, differing textual contexts lead to varying temporal granularity. In one context an event may appear atomic, while others discuss the event's substructure.

[3]The next section explains why we use a metric, point-based representation rather than Allen's interval algebra.

of events and the tightest set of temporal constraints which are directly implied by the text. As explained in the penultimate section (Experimental Results) we evaluate temporal extraction systems on both precision (correctness of temporal constraints) and recall (number of said events with reasonable bounds on their starting and ending points).

The TIE **System:** After an offline learning process, which uses TimeBank data (Pustejovsky et al. 2003b) to train TIE's probabilistic model to recognize temporal relations, TIE processes each natural language sentence in two sequential phases. First, TIE extracts events and identifies temporal expressions, applies a syntactic parser and a semantic role labeler to create a set of features. Secondly, TIE uses inference over its probabilistic model to locate inequality relations between the endpoints of the extracted elements. In summary, this paper makes the following contributions.

- We describe TIE, a novel temporal information extractor, which recogizes a wide range of temporal expressions and runs probabilistic inference to extract point-wise constraints on the endpoints of event-intervals by making full use of transitivity.

- We introduce the notion of *temporal entropy* as a means for visualizing the recall of a temporal extractor along with the tightness of the induced temporal bounds.

- We present experiments comparing TIE to three other temporal extractors, demonstrating that TIE has slightly increased precision and substantially improved recall.

Previous Work

In this section, we discuss the previous work on representations of time, methods for identifying temporal relations and existing systems for temporal information extraction.

Temporal Representation

Since Allen first proposed the interval-based algebra for representing time in natural language (Allen 1983), it has become the standard representation. However, Allen's only argument against a real-valued point-based representation is that having two intervals seamlessly *meet* requires using an "unintuitive" half-open/half-closed interval representation. On the other hand, we observe that a point approach has several benefits: 1) Reals are much simpler, only requiring standard inequalities rather than Allen's 11 interval relations, 2) In cases where quantitative interval durations are known (*e.g.*, "the war started 2 days after the incident") arithmetic fits naturally into the framework, and 3) Reasoning in Allen's framework is intractable (Vilain and Kautz 1986) whereas fast linear-programming solvers (*e.g.*, Simplex phase 1) may be used with linear inequalties.

TimeML (Verhagen et al. 2007) is a notable markup language, based roughly on Allen's relations, which is capable of annotating text to mark events, times and the links between them; an example is shown in box 2 of Figure 1. TARSQI (Verhagen et al. 2005) is useful in this regard, automatically annotating events and times.[4] TimeBank 1.2 (Pustejovsky et al. 2003b), a popular corpus of hand-annotated news articles, uses TimeML and is popular in recent research (Tatu and Srikanth 2008). The 2007 TempEval challenge (Verhagen et al. 2007) used training and test sets encoded in a subset of TimeML, which was restricted to three interval relations; entrants competed on three subtasks, each identifying a subset of temporal relations: A) between a specific event/time pair in the same sentence, B) between a specified event and the Document Creation Time, C) between the main events (provided) in two adjacent sentences. While the TempEval challenge successfully identified the strengths and weaknesses of alternative approachs to the tasks and spurred new research ideas, its three-relation representation is too simple for full temporal extraction. For example, given "A OVERLAP B" one can conclude only that $A \cap B \neq \emptyset$, but nothing about the relation between $^{\lhd}A$ and $^{\lhd}B$ or $A^{\rhd}$ and $B^{\rhd}$.

Identifying Temporal Relations

Regardless of specific details of the target temporal relations, most researchers have approached the problem of temporal identification (event/event or event/time) by using supervised learning of a classifier (Mani, Schiffman, and Zhang 2003; Lapata and Lascarides 2004; Chambers, Wang, and Jurafsky 2007; Tatu and Srikanth 2008), but this sacrifies recall by ignoring interactions due to transitivity. Yoshikawa *et al.* (Yoshikawa et al. 2009) are an exception; they use probabilistic inference to jointly solve all three TempEval tasks. Our work differs in two important ways. First, Yoshikawa's model requires Document Creation Times to enable transitivity reasoning and DCTs are unavailable for many sources on the Web, including the Wikipedia articles which are our main focus. Secondly, there is a subtle question about how much probabilistic inference a temporal extraction algorithm should do. Suppose someone asked you if the Ananji Revolution happened before 2000 and you had never heard of the event. Answering "Yes" would likely give you a good score on a TempEval-style task, because, statistically, most revolutions in history *did* happen before 2000. However, we seek to build an extraction system which uncovers all the relations most likely *directly implied by the text* not to predict the most probable relations.

Full Temporal Information Extraction

Many previous temporal information extraction systems work by directly linking events to associated time stamps (Filatova and Hovy 2001; Schilder and Habel 2001; Pasca 2008); this is a simple and practical approach, but sacrifies recall, because many events don't have an associated timestamp. By additionally considering event/event relations and applying transitivity one may derive a great many

---

[4]TARSQI is a set of tools whose functionalities include annotating events (Evita), grounding temporal expressions (GUTime), generating temporal linkings (Blinker, S2T), *etc.* When we use the name "TARSQI", we mean the temporal linking subparts of TARSQI.

Text:
In most countries of the world recovery from the Great Depression began between late 1931 and early 1933 . **1**

Annotator → In most countries of the world <EVENT eid="e133" class="OCCURRENCE" polarity="POS" pos="NOUN" tense="NONE" aspect="NONE" >**recovery**</EVENT> from the Great Depression <EVENT eid="e134" class="ASPECTUAL" polarity="POS" pos="VERB" tense="PAST" aspect="NONE" >**began**</EVENT> between late … **2**

Parser **SRL**

```
(ROOT (S  (...         3
   (NP
     (NP (NN recovery))
     (PP (IN from)
       (NP (DT the)
         (NNP Great)
         (NNP Depression))))
   (VP (VBD began)
     (PP (IN between)
       (NP
         (NP (RB late)
           (CD 1931)) ...
prep_between(began, 1931)
advmod(1933, early) ...
```

recovery	Theme (Creation) [A1] **4**
from	
the	
Great	
Depression	
began	V: begin
between	
late	
1931	temporal [AM-TMP]
and	
early	
1933	

Features →

$^\triangleleft$recovery > $^\triangleleft$1931
$^\triangleleft$recovery < 1933$^\triangleright$
$^\triangleleft$recovery > $^\triangleleft$Great Depression   **5**

…
Great Depression < 1933$^\triangleright$ **6**
…

Inference

Data labeled with temporal relations ← learn — Markov Logic Networks — exploit → Transitivity, e.g. P1 < P2 ^ P2 < P3 => P1 < P3   **0**
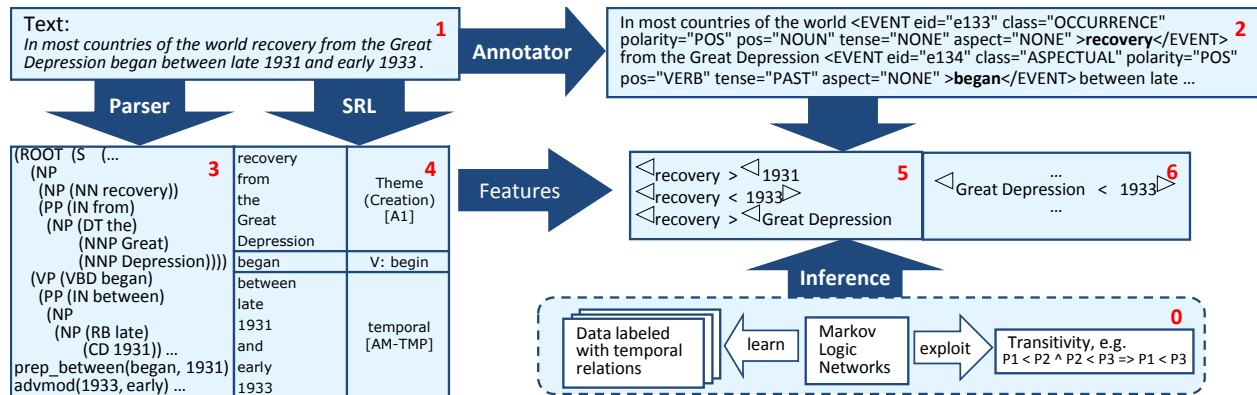
Figure 1: Diagram showing the architecture of TIE

more constraints on the times of events. In addition to transitivity reasoning, TIE extracts a wide range of events, specified using both verb phrases and dated noun phrases (*e.g.*, "The 2010 Winter Olympics had . . . "). Semantic Role Labeling systems often extract some temporal information (e.g. via the AM-TMP argument), although they don't typically reason about transitivity; we compare with Koomen's system (Koomen et al. 2005) in our experiments. Schockaert *et al.* (Schockaert, De Cock, and Kerre 2010) describe a system that takes a set of named target events as input, retrieves thousands of matching Web pages, and uses simple patterns to extract possible dates; fuzzy temporal reasoning is used to reconcile these extractions. Temporal reasoning is also required when summarizing a set of news stories, *e.g.* (Barzilay, Elhadad, and McKeown 2002), but typically it is sentences that are ordered, not events. Bramsen *et al.* (Bramsen et al. 2006) describe an algorithm for globally ordering "temporal segments" in medical case summaries.

Temporal Information Extraction

Before describing the TIE algorithm, we state the problem at hand. Given a sequence of natural language sentences, T, we seek to output a set of temporal elements E and temporal constraints C. Every element $e \in E$ should denote an event or a temporal reference (*e.g.*, "officers were *dispatched*" or "1999"). The constraints in C are linear inequalities of the form $p_1 \leq p_2 + d$, where d is a duration, often zero, and p_1 and p_2 denotes either a beginning ($^\triangleleft e$) or ending time point ($e^\triangleright$) of a temporal element, e. Our objective is to output a maximal set of events and the tightest set of temporal constraints which are directly implied by the text.

System Overview

Figure 1 summarizes the overall operation of TIE. The current TIE implementation considers each sentence $S_i \in T$ in isolation, performing the following pipeline of operations.

1. **Preprocessing:** Parse S_i using a syntactic parser (the Stanford parser (De Marneffe, MacCartney, and Manning 2006)). Detect semantic roles for verbs in the S_i by SRL (Koomen et al. 2005). Use Evita and GUTime from TARSQI (Verhagen et al. 2005) to find all temporal events and times $\{e_1, \ldots, e_n\}$ in the sentence. Generate descriptional and syntactic features for each element e_i as well as candidate point-wise relations between elements.

2. **Classification:** Using a pretrained probabilistic model (learned from TimeBank data) combined with transitivity rules, classify each pair of points (p_i, p_j) of elements the point-wise relation.

Preprocessing & Feature Assignment

We start by using Evita and GUTime (Verhagen et al. 2005) to identify the event and the time expressions in each sentence, as shown in Figure 1 box 2. We use the Stanford parser (De Marneffe, MacCartney, and Manning 2006) to create a dependency parse of the sentence (box 3). We also use a semantic role labeling system (Koomen et al. 2005) to locate the temporal arguments of verbs (box 4). At the end of preprocessing, we have generated the following features:

- **Event and Time Attributes:** In TimeML (Pustejovsky et al. 2003a), recognized events and times are associated with the attributes showing their important aspects such as tense for verb events, grounded time values etc.

- **Dependency Features:** Besides the features about each individual temporal element, it is also crucial to have good features for element pairs, (x_i, x_j). We observe from experience that most syntactic dependencies strongly indicate temporal relations. For example, in the sentence

  Australia has been `independent` since `1901`.

  the parser would output

  prep_since(`independent`, `1901`) .

The dependency prep_since (one of about 80 tokens produced by De Marneffe *et al.*'s parser) indicates that independence happens at some point in 1901 and continues to be true afterwards. TIE parses the text of each sentence to get the syntactic parse tree and the sentence's dependencies (Figure 1 box 3). Each dependency dep(w_1, w_2) is considered in turn. If w_1 and w_2 are parts of the textual expressions x_i and x_j, respectively, then TIE constructs a feature dep(x_i, x_j) to capture the relation between x_i and x_j. Statistically, these features are useful when predicting the temporal ordering between x_i and x_j.

If an event e has no dependencies to other elements, TIE creates a feature called proximity(e, x) where x is the nearest element in the parse tree. This feature avoids the situation where e cannot be linked to any element at all.

- **SRL Features:** TIE considers the AM-TMP argument (if any) for each verb identified as an event by SRL (Koomen et al. 2005). A set of MLN rules interpret the argument by recognizing the initial preposition. For example, the argument starting with "before" suggests that the verb happens before the time in the AM-TMP argument.

Identifying Temporal Relations

To identify the ordering relation for each pair of elements, we use *Markov Logic Network*s (MLN) (Richardson and Domingos 2006) to make predictions. Intuitively, Markov Logic is a probabilistic extension of first-order logic. Formally, an MLN is a set of weighted first-order formulae. Given a set of constants, a MLN can be instantiated into a ground Markov network where each node is an atom. Each formula represents a feature in the grounded Markov network with the corresponding weight. The probability of an assignment x is $P(x) = \frac{1}{Z} \exp(\sum_i w_i n_i(x))$ where Z is the normalization constant, w_i is the weight of the ith formula and $n_i(x)$ is the number of satisfied groundings for the ith formula. MLN are a flexible way to incorporate human knowledge, since they allow using different combinations of features in a straight-forward manner by setting different formula templates and then learning the weights from the training data. We use Alchemy as the implementation of MLN (Kok et al. 2005). Due to space limit, readers interested in details may refer to (Richardson and Domingos 2006).

Figure 1 box 0 illustrates how TIE's MLN formulae fit in the overall architecture. The box has a dashed outline to signify that the formula weights are learned in an offline manner (as we describe below). We use the following formula templates for relation classification:

$$\text{dep}(x, y) \rightarrow \text{after}(\text{point}(x), \text{point}(y))$$
$$\text{srl\_after}(p, q) \rightarrow \text{after}(p, q)$$
$$\text{after}(p, q) \wedge \text{after}(q, r) \rightarrow \text{after}(p, q)$$

The first formula models the influence of each syntactic dependency on the temporal relation between the arguments; this formula is actually second-order, because dep stands for proximity or any of the ~ 80 Stanford dependencies; point denotes either the starting or ending point of the element. The second formula integrates temporal information provided by the SRL system. The last MLN rule decreases the probability of interpretations which are inconsistent with transitivity.

Learning The weights for these formulae are learned from a selected portion of the TimeBank data set.[5] Each Allen-style relation is translated into temporal constraints in terms of its beginning and ending points (e.g. Figure 1 box 5). We filter the data by selecting the relations whose arguments reside in the same sentence, since TimeBank also has labels for inter-sentence pairs of events.

Inference For each sentence, we run inference using MC-SAT (Poon and Domingos 2006) with default param-

eters and return marginal probabilities for the relations of all possible pairs. All temporal relations whose probability is over threshold are returned as TIE's predications. Varying the threshold moves TIE along the precision/recall curve.

Experimentation

This section addresses following questions: 1) How accurately can TIE perform relation classification? 2) How well can it bound the times of events? and 3) What factors are important to TIE's performance?

Data Set The data for our experiments[6] were collected from Wikipedia articles[7]. We selected the four representative categories of articles (the number of articles is in the brackets): Warfare(63), Universities(59), Celebrities(99) and Riots(262). Summing over all articles yields a total 40640 sentences. We randomly sampled 45 sentences for hand-labeling prior to testing. Within this test set, there are 151 events, 56 times identified by GUTime and Evita and therefore 644 potential constraints between one point of an event and the other point of a time for testing.

To get ground truth, we asked people to label the temporal relations between all pairs of elements $\{(x_i, x_j)\}$ by comparing their start and ending points. Each pair was labeled by at least two people. If the assigned labels did not agree, a third person resolved the disagreement.

Methods We present two evalution metrics. One is the familiar precision and recall. With the gold constraints, these scores are computed by comparing with the constraints each system outputs. Precision is computed as $P = \sum_i c_i / \sum_i p_i$ where c_i is the number of correctly predicted constraints for i^{th} sentence, p_i is the number of predictions and the summation is over all test data.

Note, however, that recall is an inadquate metric for temporal IE, since it records only whether an ordering relation has been extracted, but doesn't measure how *tightly* an event is bounded in time. To compare the degree to which different systems constrain the time of events, we introduce a novel "recall-like" measure, *Temporal Entropy* (TE), for evaluating the number and "tightness" of induced temporal bounds. When a human comprehends a piece of text, she can give time bounds to the starting and end points of the events in the text. We take the logarithm (base 10) of the length of the tightest bound on a time point as its TE. Similarly, we may compute and compare the logarithm of system-derived bounds for the endpoints of each event. If a system fails to *recognize* an event as such, we assign it a maximum entropy equal to the log of the maximal temporal duration plus two.[8] We visualize TE on the y axis of a graph where the x axis represents different time points, sorted so that those with the least entropy have the lowest index.

Extraction Algorithms Compared We implemented three additional extractors for comparison with TIE. The

---

[5]In practice, we manually gave positive weights to the SRL and transitivity rules to save training time. We also impose a negative prior to the after predicate to suppress predictions unless there is substantial evidence.

[8]The addition of "two" is arbitrary; we choose this value for the plots in order to graphically separate the points where a system find no bounds (the gray area in the upper part of Figure 3 & 4) from events where the system bounded weakly.

first (PASCA) is a pattern-based extraction system for question answering, which we reimplemented following (Pasca 2008). It exploits four lexico-syntactic patterns to find the temporal expressions, t, in each sentence, such as sentences that start or end with a simple adverbial phrase containing a date. Associated events, e, were assigned to the major verb. By inspecting the patterns, we arrived at the appropriate constraints: $^{\triangleleft}t < {^{\triangleleft}e} < e^{\triangleright} < t^{\triangleright}$.

The second baseline is the SRL system (Koomen et al. 2005) described in the previous section. We use TARSQI as the third system with point-based constraints translated from the Allen-style relations. For fairness, we performed constraint propagation (Vilain and Kautz 1986) for all three systems to get the final results.
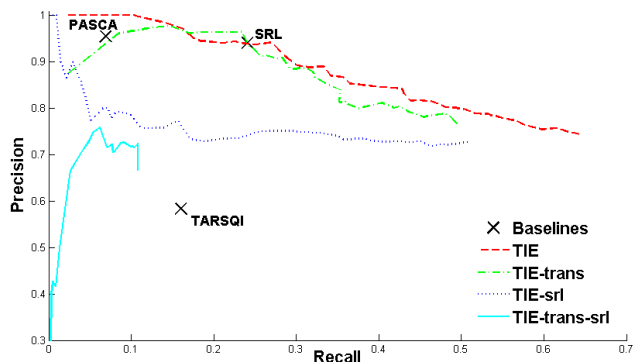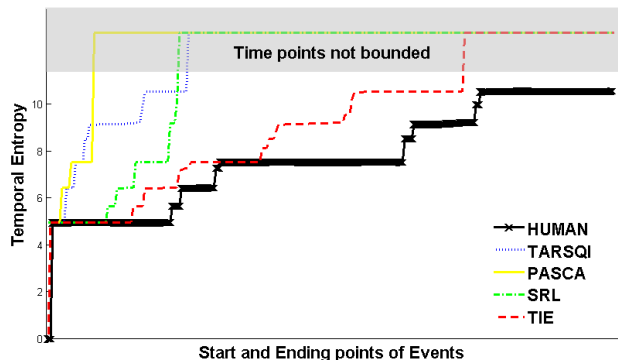

Figure 2: Precision-Recall curve.

Figure 3: Temporal Entropy. The curve of TIE is the closest to the gold curve (HUMAN).

Results Figure 2 depicts the precision-recall curve of TIE by enumerating its probability threshold from 0.50 to 0.99. As seen from the curve, TIE is capable of very accurate extractions (the upper left corner of the plot). Compared to PASCA, TIE extracts more correct constraints at comparable precision. However, we note that Pasca's system runs much more quickly, taking only few milliseconds per sentence, while TIE takes a few seconds depending on the number of temporal elements in the sentence. TIE is comparable with SRL at 0.95 precision. Also, without sacrificing much precision, TIE is able to triple recall.

In Figure 3[9], the temporal entropies are ordered increas-

---

[9]We chose 0.5 as the threshold for displaying the temporal entropy plot throughout the experiments.

ingly. A lower curve means reduced entropy and better performance. TIE (the dashed line) is closest to a human's skill.
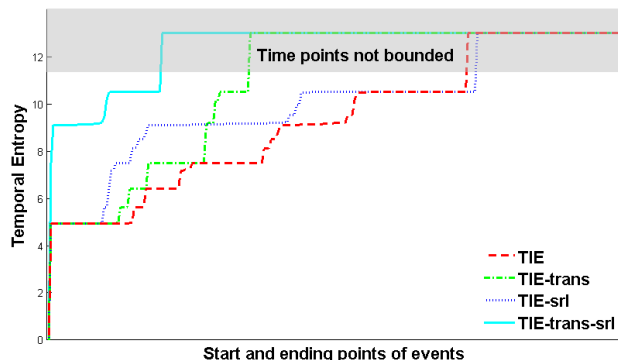

Figure 4: Temporal Entropy. Ablation Test.

Ablation Test To understand what factors are crucial to TIE's performance, we removed some features from the system. The results are shown in Figure 2 and Figure 4. First, in order to see the importance of SRL features in a Temporal IE system. We removed the SRL features in TIE-srl. Secondly, we are also interested in investigating how much the transitivity helps bound the events. To achieve that goal, we removed the transitivity rule from our model (TIE-trans). TIE-trans-srl omits both features. Let us take a look at the temporal entropy plot (Figure 4) in two orthogonal directions. Horizontally, we can see how many points the system is able to bound, i.e. recall. Vertically, the area between the curves tells the difference in tightness of temporal bounds; in other words, the lower the curve is, the tighter bounds the system gives. In the experiments we observed that SRL features improve precision, and complementarily, the transitivity rule helps increase the recall.

TempEval Task A Although TIE was not designed to perform the TempEval tasks, does not use information about Document Creation Times (DCTs), and produces more granular results, we report on it's performance for completeness. TIE's point-wise constraints from TIE's output are converted back to Allen-style relations. The data set for testing is further restricted to {BEFORE, AFTER, OVERLAP} (from 169 to 148). We see that TIE's accuracy is 0.695, which is comparable to 0.716 by the state-of-the-art algorithm (Yoshikawa et al. 2009)[10].

Conclusion

In this paper, we describe TIE, a novel temporal information extractor, that uses probabilistic inference to extract point-wise constraints on the endpoints of event-intervals by taking advantage of transitivity. Secondly, we introduce the notion of *temporal entropy* as a means for visualizing the recall of a temporal extractor along with the tightness of the induced temporal bounds. Third, we present experiments comparing TIE to three other temporal extractors, demonstrating that TIE outperforms other systems by a large margin. In the future we hope to extend TIE in several ways, including incorporation of inter-sentence event coreference and point-wise constraints with more than two arguments.

---

[10]The accuracy is adjusted to the smaller test set.

References

Agichtein, E., and Gravano, L. 2000. Snowball: Extracting Relations from Large Plain-Text Collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, 85–94.

Allen, J. 1983. Maintaining knowledge about temporal intervals. *C. ACM* 26(11):832–843.

Banko, M.; Cafarella, M.; Soderland, S.; Broadhead, M.; and Etzioni, O. 2007. Open information extraction from the Web. In *Procs. of IJCAI*.

Barzilay, R.; Elhadad, N.; and McKeown, K. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *JAIR* 17:35–55.

Bramsen, P.; Deshpande, P.; Lee, Y.; and Barzilay, R. 2006. Inducing temporal graphs. In *Procs. of EMNLP*, 189–198.

Brin, S. 1998. Extracting Patterns and Relations from the World Wide Web. In *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT'98*, 172–183.

Chambers, N.; Wang, S.; and Jurafsky, D. 2007. Classifying temporal relations between events. In *ACL*, 173–176.

Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; and Slattery, S. 2000. Learning to Construct Knowledge Bases from the World Wide Web. *Artificial Intelligence 118(1-2)* 69–113.

De Marneffe, M.; MacCartney, B.; and Manning, C. 2006. Generating typed dependency parses from phrase structure parses. In *LREC 2006*.

Etzioni, O.; Cafarella, M.; Downey, D.; Kok, S.; Popescu, A.; Shaked, T.; Soderland, S.; Weld, D.; and Yates, A. 2005. Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence* 165(1):91–134.

Filatova, E., and Hovy, E. 2001. Assigning time-stamps to event-clauses. In *Proceedings of ACL Workshop on Temporal and Spatial Reasoning*, 88–95.

Kok, S.; Singla, P.; Richardson, M.; Domingos, P.; Sumner, M.; Poon, H.; and Lowd, D. 2005. The Alchemy system for statistical relational AI. *Dept. of Computer Science and Engineering, University of Washington, Technical Report* 2(6).

Koomen, P.; Punyakanok, V.; Roth, D.; and Yih, W. 2005. Generalized inference with multiple semantic role labeling systems (shared task paper). In *Proc. of CoNLL*.

Lapata, M., and Lascarides, A. 2004. Inferring sentence-internal temporal relations. In *Procs. of HLT-NAACL*.

Mani, I.; Schiffman, B.; and Zhang, J. 2003. Inferring temporal ordering of events in news. In *Procs. of HLT-NAACL*.

Pasca, M. 2008. Answering Definition Questions via Temporally-Anchored Text Snippets. *Proc. of IJCNLP2008*.

Poon, H., and Domingos, P. 2006. Sound and efficient inference with probabilistic and deterministic dependencies. In *Proceedings of AAAI*.

Pustejovsky, J.; Castano, J.; Ingria, R.; Saurí, R.; Gaizauskas, R.; Setzer, A.; Katz, G.; and Radev, D. 2003a. TimeML: Robust specification of event and temporal expressions in text. In *IWCS-5 Fifth Intl. Workshop on Computational Semantics*.

Pustejovsky, J.; Hanks, P.; Sauri, R.; See, A.; Gaizauskas, R.; Setzer, A.; Radev, D.; Sundheim, B.; Day, D.; Ferro, L.; et al. 2003b. The timebank corpus. In *Corpus Linguistics*.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62(1):107–136.

Riloff, E., and Jones, R. 1999. Learning Dictionaries for Information Extraction by Multi-level Bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 474–479.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach, 2nd ed.* Prentice Hall.

Schilder, F., and Habel, C. 2001. From temporal expressions to temporal information: Semantic tagging of news messages. In *Proceedings of the ACL-2001 Workshop on Temporal and Spatial Information Processing*, 65–72.

Schockaert, S.; De Cock, M.; and Kerre, E. 2010. Reasoning about fuzzy temporal information from the web: towards retrieval of historical events. *Soft Computing: A Fusion of Foundations, Methodologies and Applications* 14(8).

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: A core of semantic knowledge - unifying WordNet and Wikipedia. In *Proceedings of WWW07*.

Tatu, M., and Srikanth, M. 2008. Experiments with reasoning for temporal relations between events. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*.

Verhagen, M.; Mani, I.; Saurí, R.; Knippen, R.; Littman, J.; and Pustejovsky, J. 2005. Automating Temporal Annotation with TARSQI. In *Demo Session. Proceedings of ACL*.

Verhagen, M.; Gaizauskas, R.; Schilder, F.; Hepple, M.; Katz, G.; and Pustejovsky, J. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *SemEval-2007: 4th International Workshop on Semantic Evaluations*.

Vilain, M., and Kautz, H. 1986. Constraint propagation algorithms for temporal reasoning. In *Proceedings of the Fifth National Conference on Artificial Intelligence*.

Wu, F., and Weld, D. 2007. Autonomously semantifying Wikipedia. In *Proceedings of the ACM Sixteenth Conference on Information and Knowledge Management (CIKM-07)*.

Yoshikawa, K.; NAIST, J.; Riedel, S.; Asahara, M.; and Matsumoto, Y. 2009. Jointly Identifying Temporal Relations with Markov Logic. In *Proceedings of the 2009 ACL*.