Evita: A Robust Event Recognizer For QA Systems

Roser Saurí   **Robert Knippen**   **Marc Verhagen**   **James Pustejovsky**

Lab for Linguistics and Computation
Computer Science Department
Brandeis University
415 South Street, Waltham, MA 02454, USA
`{roser,knippen,marc,jamesp}@cs.brandeis.edu`

Abstract

We present *Evita*, an application for recognizing events in natural language texts. Although developed as part of a suite of tools aimed at providing question answering systems with information about both temporal and intensional relations among events, it can be used independently as an event extraction tool. It is unique in that it is not limited to any pre-established list of relation types (events), nor is it restricted to a specific domain. Evita performs the identification and tagging of event expressions based on fairly simple strategies, informed by both linguistic- and statistically-based data. It achieves a performance ratio of 80.12% F-measure.[1]

1 Introduction

Event recognition is, after entity recognition, one of the major tasks within Information Extraction. It is currently being succesfully applied in different areas, like bioinformatics and text classification. Recognizing events in these fields is generally carried out by means of pre-defined sets of relations, possibly structured into an ontology, which makes such tasks domain dependent, but feasible. Event recognition is also at the core of Question Answering,

since input questions touch on events and situations in the world (states, actions, properties, etc.), as they are reported in the text. In this field as well, the use of pre-defined sets of relation patterns has proved fairly reliable, particularly in the case of factoid type queries (Brill et al., 2002; Ravichandran and Hovy, 2002; Hovy et al., 2002; Soubbotin and Soubbotin, 2002).

Nonetheless, such an approach is not sensitive to certain contextual elements that may be fundamental for returning the appropriate answer. This is for instance the case in reporting or attempting contexts. Given the passage in (1a), a pattern-generated answer to question (1b) would be (1c). Similarly, disregarding the reporting context in example (2) could erroneously lead to concluding that no one from the White House was involved in the Watergate affair.

(1) a. Of the 14 known ways to reach the summit, only the East Ridge route has never been successfully climbed since George Mallory and Andrew "Sandy" Irvine first attempted to climb Everest in 1924.

   b. When did George Mallory and Andrew Irvine first climb Everest?

   c. #In 1924.

(2) a. Nixon claimed that White House counsel John Dean had conducted an investigation into the Watergate matter and found that no-one from the White House was involved.

   b. What members of the White House were involved in the Watergate matter?

   c. #Nobody.

Intensional contexts like those above are generated by predicates referring to events of attempting, intending, commanding, and reporting, among others. When present in text, they function as modal

qualifiers of the truth of a given proposition, as in example (2), or they indicate the factuality nature of the event expressed by the proposition (whether it happened or not), as in (1) (Saurí and Verhagen, 2005).

The need for a more sophisticated approach that sheds some awareness on the specificity of certain linguistic contexts is in line with the results obtained in previous TREC Question Answering competitions (Voorhees, 2002, 2003). There, a system that attempted a minimal understanding of both the question and the answer candidates, by translating them into their logical forms and using an inference engine, achieved a notably higher score than any surface-based system (Moldavan et al., 2002; Harabagiu et al., 2003).

Non-factoid questions introduce an even higher level of difficulty. Unlike factoid questions, there is no simple or unique answer, but more or less satisfactory ones instead. In many cases, they involve dealing with several events, or identifying and reasoning about certain relations among events which are only partially stated in the source documents (such as temporal and causal ones), all of which makes the pattern-based approach less suitable for the task (Small et al., 2003, Soricut and Brill, 2004). Temporal information in particular plays a significant role in the context of question answering systems (Pustejovsky et al., forthcoming). The question in (3), for instance, requires identifying a set of events related to the referred killing of peasants in Mexico, and subsequently ordering them along a temporal axis.

(3) What happened in Chiapas, Mexico, after the killing of 45 peasants in Acteal?

Reasoning about events in intensional contexts, or with event-ordering relations such as temporality and causality, is a requisite for any open-domain QA system aiming at both factoid and non-factoid questions. As a first step, this involves the identification of all relevant events reported in the source documents, so that later processing stages can locate intensional context boundaries and temporal relations among these events.

In this article, we present **Evita**, a tool for recognizing events in natural language texts. It has been developed as part of a suite of tools aimed at providing QA systems with information about both temporal and intensional relations between events; we anticipate, however, that it will be useful for other NLP tasks as well, such as narrative understanding, summarization, and the creation of factual databases from textual sources.

In the next section, we provide the linguistic foundations and technical details of our event recognizer tool. Section 3 gives the results and discusses them in the context of the task. We conclude in section 4, with an overview of Evita's main achievements and a brief discussion of future directions.

2 Evita, An Event Recognition Tool

Evita ('<u>E</u>vents <u>I</u>n <u>T</u>ext <u>A</u>nalyzer') is an event recognition system developed under the ARDA-funded TARSQI research framework. TARSQI is devoted to two complementary lines of work: (1) establishing a specification language, TimeML, aimed at capturing the richness of temporal and event related information in language (Pustejovsky et al., 2003a, forthcoming), and (2) the construction of a set of tools that perform tasks of identifying, tagging, and reasoning about eventive and temporal information in natural language texts (Pustejovsky and Gaizauskas, forthcoming, Mani, 2005; Mani and Schiffman, forthcoming; Verhagen, 2004; Verhagen et al., 2005; Verhagen and Knippen, forthcoming). Within TARSQI's framework, Evita's role is locating and tagging all event-referring expressions in the input text that can be temporally ordered.

Evita combines linguistic- and statistically-based techniques to better address all subtasks of event recognition. For example, the module devoted to recognizing temporal information that is expressed through the morphology of certain event expressions (such as tense and aspect) uses grammatical information (see section 2.4), whereas disambiguating nouns that can have both eventive and non-eventive interpretations is carried out by a statistical module (section 2.3).

The functionality of Evita breaks down into two parts: event identification and analysis of the event-based grammatical features that are relevant for temporal reasoning purposes. Both tasks rely on a preprocessing step which performs part-of-speech tag-

ging and chunking, and on a module for clustering together chunks that refer to the same event. In the following subsection we provide the linguistic assumptions informing Evita. Then, subsections 2.2 to 2.5 provide a detailed description of Evita's different subcomponents: preprocessing, clustering of chunks, event identification, and analysis of the grammatical features associated to events.

2.1 Linguistic settings

TimeML identifies as events those event-denoting expressions that participate in the narrative of a given document and which can be temporally ordered. This includes all dynamic situations (punctual or durative) that happen or occur in the text, but also states in which something obtains or holds true, if they are temporally located in the text. As a result, generics and most state-denoting expressions are filtered out (see Saurí et al. (2004) for a more exhaustive definition of the criteria for event candidacy in TimeML).

Event-denoting expressions are found in a wide range of syntactic expressions, such as finite clauses (*that no-one from the White House* **was involved**), nonfinite clauses (*to* **climb** *Everest*), noun phrases headed by nominalizations (*the young industry's rapid* **growth***, several anti-war* **demonstrations**) or event-referring nouns (*the controversial* **war**), and adjective phrases (*fully* **prepared**).

In addition to identifying the textual extent of events, Evita also analyzes certain grammatical features associated with them. These include:

- The *polarity* (positive or negative) of the expression tells whether the referred event has happened or not;

- *Modality* (as marked by modal auxiliaries *may, can, might, could, should*, etc., or adverbials like *probably, likely*, etc.) qualifies the denoted event with modal information (irrealis, necessity, possibility), and therefore has implications for the suitability of statements as answers to questions, in a parallel way to other intensional contexts exemplified in (1-2);

- *Tense* and *aspect* provide crucial information for the temporal ordering of the events;

- Similarly, the *non-finite morphology* of certain verbal expressions (infinitival, present partici-

ple, or past participle) has been shown as useful in predicting temporal relations between events (Lapata and Lascarides, 2004). We also consider as possible values here the categories of noun and adjective.

- Event *class* distinguishes among states (e.g., *be the director of*), general occurrences (*walk*), reporting (*tell*), intensional (*attempt*), and perception (*observe*) events. This classification is relevant for characterizing the nature of the event as irrealis, factual, possible, reported, etc. (recall examples (1-2) above).

Despite the fact that modality, tense, aspect, and non-finite morphology are typically verbal features, some nouns and adjectives can also have this sort of information associated with them; in particular, when they are part of the predicative complement of a copular verb (e.g., *may be* **ready***, had been a* **collaborator**). A TimeML mark-up of these cases will tag only the complement as an event, disregarding the copular verb. Therefore, the modality, tense, aspect, and non-finite morphology information associated with the verb is incorporated as part of the event identified as the nominal or adjectival complement.

Except for *event class*, the characterization of all the features above relies strictly on surface linguistic cues. Notice that this surface-based approach does not provide for the actual temporal interpretation of the events in the given context. The tense of a verbal phrase, for example, does not always map in a straightforward way with the time being referred to in the world; e.g., simple present is sometimes used to express future time or habituality. We handle the task of mapping event features onto their semantics during a later processing stage, not addressed in this paper, but see Mani and Schiffman (forthcoming).

TimeML does not identify event participants, but the event tag and its attributes have been designed to interface with Named Entity taggers in a straightforward manner. In fact, the issue of argument linking to the events in TimeML is already being addressed in the effort to create a unified annotation with PropBank and NomBank (Pustejovsky et al. 2005). A complete overview of the linguistic foundations of TimeML can be obtained in Pustejovsky et al. (forthcoming).

2.2 Preprocessing

For the task of event recognition, Evita needs access to part of speech tags and to the result of some form of syntactic parsing. Section 2.1 above detailed some of the different syntactic structures that are used to refer to events. However, using a shallow parser is enough to retrieve event referring expressions, since they are generally conveyed by three possible part of speech categories: verbs (*go, see, say*), nouns (*departure, glimpse, war*), and adjectives (*upset, pregnant, dead*).

Part of speech tags and phrase chunks are also valuable for the identification of certain grammatical features such as tense, non-finite morphology, or polarity. Finally, lexical stems are necessary for those tasks involving lexical look-up. We obtain all such grammatical information by first preprocessing the input file using the Alembic Workbench tagger, lemmatizer, and chunker (Day et al., 1997). Evita's input must be XML-compliant, but need not conform to the TimeML DTD.

2.3 Event Recognition

Event identification in Evita is based on the notion of event as defined in the previous section. Only lexical items tagged by the preprocessing stage as either verbs, nouns, or adjectives are considered event candidates.

Different strategies are used for identifying events in these three categories. Event identification in verbal chunks is based on lexical look-up, accompanied by minimal contextual parsing in order to exclude weak stative predicates, such as 'be', and some generics (e.g., verbs with bare plural subjects). For every verbal chunk in the text, Evita first applies a pattern-based selection step that distinguishes among different kinds of information: the chunk head, which is generally the most-right element of verbal nature in the chunk, thus disregarding particles of different sort and punctuation marks; the modal auxiliary sequence, if any (e.g., *may have to*); the sequence of *do*, *have*, or *be* auxiliaries, marking for aspect, tense and voice; and finally, any item expressing the polarity of the event. The last three pieces of information will be used later, when identifying the event grammatical features (section 2.4).

Based on basic lexical inventories, the chunk may then be rejected if the head belongs to a certain class. For instance, copular verbs are generally disregarded for event tagging, although they enter into a a process of chunk clustering, together with their predicative complement (see section 2.5).

The identification of nominal and adjectival events is also initiated by the step of information selection. For each noun and adjective chunk, their head and polarity markers, if any, are distinguished.

Identifying events expressed by nouns involves two parts: a phase of lexical lookup, and a disambiguation process. The lexical lookup aims at an initial filtering of candidates to nominal events. First, Evita checks whether the head of the noun chunk is an event in WordNet. We identified about 25 subtrees from WordNet where all synsets denote nominal events. One of these, the largest, is the tree underneath the synset that contains the word *event*. Other subtrees were selected by analyzing events in SemCor and TimeBank1.2[2] and mapping them to WordNet synsets. One example is the synset with the noun *phenomenon*. In some cases, exceptions are defined. For example, a noun in a subset subsumed by the *phenomenon* synset is not an event if it is also subsumed by the synset with the noun *cloud* (in other words, many phenomena are events but clouds are not).

If the result of lexical lookup is inconclusive (that is, if a nominal occurs in WN as both and event and a non-event), then a disambiguation step is applied. This process is based on rules learned by a Bayesian classifier trained on SemCor.

Finally, identifying events from adjectives takes a conservative approach of tagging as events only those adjectives that were annotated as such in TimeBank1.2, whenever they appear as the head of a predicative complement. Thus, in addition to the use of corpus-based data, the subtask relies again on a minimal contextual parsing capable of identifying the complements of copular predicates.

2.4 Identification of Grammatical Features

Identifying the grammatical features of events follows different procedures, depending on the part of speech of the event-denoting expression, and whether the feature is explicitly realized by the morphology of such expressions.

In event-denoting expressions that contain a verbal chunk, *tense*, *aspect*, and *non-finite morphology* values are directly derivable from the morphology of this constituent, which in English is quite straightforward. Thus, the identification of these features is done by first extracting the verbal constituents from the verbal chunk (disregarding adverbials, punctuation marks, etc.), and then applying a set of over 140 simple linguistic rules, which define different possible verbal phrases and map them to their corresponding tense, aspect, and non-finite morphology values. Figure 1 illustrates the rule for verbal phrases of future tense, progressive aspect, which bear the modal form *have to* (as in, e.g., *Participants* **will have to be working** *on the same topics*):

```
[form in futureForm],
[form=='have'],
[form=='to', pos=='TO'],
[form=='be'], [pos=='VBG'],
==>
[tense='FUTURE',
aspect='PROGRESSIVE',
nf_morph='NONE']
```

Figure 1: Grammatical Rule

For event-denoting expressions containing no verbal chunk, tense and aspect is established as null ('NONE' value), and non-finite morphology is 'noun' or 'adjective', depending on the part-of-speech of their head.

Modality and *polarity* are the two remaining morphology-based features identified here. Evita extracts the values of these two attributes using basic pattern-matching techniques over the approapriate verbal, nominal, or adjectival chunk.

On the other hand, the identification of event *class* cannot rely on linguistic cues such as the morphology of the expression. Instead, it requires a combination of lexical resource-based look-up and word sense disambiguation. At present, this task has been attempted only in a very preliminary way, by tagging events with the class that was most frequently as-

signed to them in TimeBank1.2. Despite the limitations of such a treatment, the accuracy ratio is fairly good (refer to section 3).

2.5 Clustering of Chunks

In some cases, the chunker applied at the preprocessing stage identifies two independent constituents that contribute information about the same event. This may be due to a chunker error, but it is also systematically the case in verbal phrases containing the *have to* modal form or the *be going to* future form (Figure 2).

```
<VG>
<VX><lex pos="VBD">had</lex></VX>
</VG>
<VG-INF>
<INF><lex pos="TO">to</lex>
<lex pos="VB">say</lex>
</INF>

</VG-INF>
```

Figure 2: *have to* VP

It may be also necessary in verbal phrases with other modal auxiliaries, or with auxiliary forms of the *have*, *do*, or *be* forms, in which the auxiliary part is split off the main verb because of the presence of an adverbial phrase or similar (Figure 3).

```
<VG>
<VX><lex pos="VBZ">has</lex></VX>
</VG>
<lex pos=",">,</lex>
<lex pos="IN">of</lex>
<NG>
<HEAD><lex pos="NN">course</lex></HEAD>
</NG>
<lex pos=",">,</lex>
<VG>
<VX><lex pos="VBD">tried</lex></VX>
</VG>
```

Figure 3: *have V_en* VP

Constructions with copular verbs are another kind of context which requires clustering of chunks, in order to group together the verbal chunk corresponding to the copular predicate and the non-verbal chunk that functions as its predicative complement. In all these cases, additional syntactic parsing is needed for the tasks of event recognition and grammatical feature identification, in order to cluster together the two independent chunks.

The task of clustering chunks into bigger ones is activated by specific triggers (e.g., a chunk headed by an auxiliary form, or a chunk headed by the copular verb *be*) and carried out locally in the context of that trigger. For each trigger, there is a set of grammatical patterns describing the possible structures it can be a constituent of. The form *have*, for instance, may be followed by an infinitival phrase *to V*, constituting part of the modal form *have to* in the bigger verbal group *have to V*, as in Figure 2 above, or it may also be followed by a past participle-headed chunk, with which it forms a bigger verbal phrase *have V-en* expressing perfective aspect (Figure 3).

The grammatical patterns established for each trigger are written using the standard syntax of regular expressions, allowing for a greater expressiveness in the description of sequences of chunks (optionality of elements, inclusion of adverbial phrases and punctuation marks, variability in length, etc.). These patterns are then compiled into finite state automata that work with grammatical objects instead of string characters. Such an approach is based on well-established techniques using finite-state methods (see for instance Koskenniemi, 1992; Appelt et al. 1993; Karttunen et al., 1996; Grefenstette, 1996, among others).

Evita sequentially feeds each of the FSAs for the current trigger with the right-side part of the trigger context (up to the first sentence boundary), which is represented as a sequence of grammatical objects. If one of the FSAs accepts this sequence or a subpart of it, then the clustering operation is applied on the chunks within the accepted (sub)sequence.

3 Results

Evaluation of Evita has been carried out by comparing its performance against TimeBank1.2. The current performance of Evita is at 74.03% precision, 87.31% recall, for a resulting F-measure of 80.12% (with β=0.5). These results are comparable to the interannotation agreement scores for the task of tagging verbal and nominal events, by graduate linguistics students with only basic training (Table 1).[3] By basic training we understand that they had read

the guidelines, had been given some additional advice, and subsequently annotated over 10 documents before annotating those used in the interannotation evaluation. They did not, however, have any meetings amongst themselves in order to discuss issues or to agree on a common strategy.

Category	F-measure
Nouns	64%
Verbs	80%

Table 1: Interannotation Agreement

On the other hand, the Accuracy ratio (i.e., the percentage of values Evita marked according to the gold standard) on the identification of event grammatical features is as shown:

Feature	Accuracy
polarity	98.26%
aspect	97.87%
modality	97.02%
tense	92.05%
nf_morph	89.95%
class	86.26%

Table 2: Accuracy of Grammatical Features

Accuracy for *polarity, aspect*, and *modality* is optimal: over 97% in all three cases. In fact, we were expecting a lower accuracy for polarity, since Evita relies only on the polarity elements present in the chunk containg the event, but does not take into account non-local forms of expressing polarity in English, such as negative polarity on the subject of a sentence (as in *Nobody saw him* or in *No victims were found*).

The slightly lower ratio for *tense* and *nf_morph* is in most of the cases due to problems from the POS tagger used in the preprocessing step, since tense and non-finite morphology values are mainly based on its result. Some common POS tagging mistakes deriving on tense and nf_morph errors are, for instance, identifying a present form as the base form of the verb, a simple past form as a past participle form, or vice versa. Errors in the nf_morph value are also due to the difficulty in distinguishing sometimes between present participle and noun (for *ing*-forms), or between past participle and adjective.

---

[3]These figures are also in terms of F-measure. See Hripcsak and Rothschild (2005) for the use of such metric in order to quantify interannotator reliability.

The lowest score is for event *class*, which nevertheless is in the 80s%. This is the only feature that cannot be obtained based on surface cues. Evita's treatment of this feature is still very basic, and we envision that it can be easily enhanced by exploring standard word sense disambiguation techniques.

4 Discussion and Conclusions

We have presented Evita, a tool for recognizing and tagging events in natural language text. To our knowledge, this is a unique tool within the community, in that it is not based on any pre-established list of event patterns, nor is it restricted to a specific domain. In addition, Evita identifies the grammatical information that is associated with the event-referring expression, such as tense, aspect, polarity, and modality. The characterization of these features is based on explicit linguistic cues. Unlike other work on event recognition, Evita does not attempt to identify event participants, but relies on the use of entity taggers for the linking of arguments to events.

Evita combines linguistic- and statistically-based knowledge to better address each particular subtask of the event recognition problem. Linguistic knowledge has been used for the parsing of very local and controlled contexts, such as verbal phrases, and the extraction of morphologically explicit information. On the other hand, statistical knowledge has contributed to the process of disambiguation of nominal events, following the current trend in the Word Sense Disambiguation field.

Our tool is grounded on simple and well-known technologies; namely, a standard preprocessing stage, finite state techniques, and Bayesian-based techniques for word sense disambiguation. In addition, it is conceived from a highly modular perspective. Thus, an effort has been put on separating linguistic knowledge from the processing thread. In this way we guarantee a low-cost maintainance of the system, and simplify the task of enriching the grammatical knowledge (which can be carried out even by naive programmers such as linguists) when additional data is obtained from corpus exploitation.

Evita is a component within a larger suite of tools. It is one of the steps within a processing sequence which aims at providing basic semantic information (such as temporal relations or intensional context boundaries) to applications like Question Answering or Narrative Understanding, for which text understanding is shown to be fundamental, in addition to shallow-based techniques. Nonetheless, Evita can also be used independently for purposes other than those above.

Additional tools within the TimeML research framework are (a) GUTime, a recognizer of temporal expressions which extends Tempex for TimeML (Mani, 2005), (b) a tool devoted to the temporal ordering and anchoring of events (Mani and Schiffman, forthcoming), and (c) Slinket, an application in charge of identifying subordination contexts that introduce intensional events like those exemplified in (1-2) (Verhagen et al., 2005). Together with these, Evita provides capabilities for a more adequate treatment of temporal and intensional information in textual sources, thereby contributing towards incorporating greater inferential capabilities to applications within QA and related fields, a requisite that has been shown necessary in the Introduction section.

Further work on Evita will be focused on two main areas: (1) improving the sense disambiguation of candidates to event nominals by experimenting with additional learning techniques, and (2) improving event classification. The accuracy ratio for this latter task is already fairly acceptable (86.26%), but it still needs to be enhanced in order to guarantee an optimal detection of subordinating intensional contexts (recall examples 1-2). Both lines of work will involve the exploration and use of word sense disambiguation techniques.

References

Appelt, Douglas E., Jerry R. Hobbs, John Bear, David Israel and Mabry Tyson 1993. 'FASTUS: A Finite-state Processor for Information Extraction from Real-world Text'. *Proceedings IJCAI-93*.

Brill, Eric, Susan Dumais and Michele Banko. 2002. 'An Analysis of the AskMSR Question Answering System'. *Proceedings of EMNLP 2002*.

Day, David,, John Aberdeen, Lynette Hirschman, Robyn Kozierok, Patricia Robinson and Marc Vilain. 1997. 'Mixed-Initiative Development of Language Processing Systems'. *Fifth Conference on Applied Natural Language Processing Systems*: 88–95.

Grefenstette, Gregory. 1996. 'Light Parsing as Finite-State Filtering'. *Workshop on Extended Finite State Models of Language, ECAI'96*.

Harabagiu, S., D. Moldovan, C. Clark, M. Bowden, J. Williams and J. Bensley. 2003. 'Answer Mining by Combining Extraction Techniques with Abductive Reasoning'. *Proceedings of the Text Retrieval Conference, TREC 2003*: 375-382.

Hovy, Eduard, Ulf Hermjakob and Deepak Ravichandran. 2002. A Question/Answer Typology with Surface Text Patterns. *Proceedings of the Second International Conference on Human Language Technology Research, HLT 2002*: 247-251.

Hripcsak, George and Adam S. Rothschild. 2005. 'Agreement, the F-measure, and reliability in information retrieval'. *Journal of the American Medical Informatics Association*, 12: 296-298.

Karttunen, L., J-P. Chanod, G. Grefenstette and A. Schiller. 1996. 'Regular Expressions for Language Engineering'. *Natural Language Engineering*, 2(4).

Koskenniemi, Kimmo, Pasi Tapanainen and Atro Voutilainen. 'Compiling and Using Finite-State Syntactic Rules'. *Proceedings of COLING-92*: 156-162.

Lapata, Maria and Alex Lascarides 2004. Inferring Sentence-Internal Temporal Relations. *Proceedings of HLT-NAACL 2004*.

Mani, Inderjeet. 2005. *Time Expression Tagger and Normalizer*. http://complingone.georgetown.edu/ linguist/GU_TIME_DOWNLOAD.HTML

Mani, Inderjeet and Barry Schiffman. Forthcoming. 'Temporally Anchoring and Ordering Events in News'. James Pustejovsky and Robert Gaizauskas (eds.) *Event Recognition in Natural Language*. John Benjamins.

Moldovan, D., S. Harabagiu, R. Girju, P. Morarescu, F. Lacatusu, A. Novischi, A. Badulescu and O. Bolohan. 2002. 'LCC Tools for Question Answering'. *Proceedings of the Text REtrieval Conference, TREC 2002*.

Pustejovsky, J., J. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, and G. Katz. 2003a. TimeML: Robust Specification of Event and Temporal Expressions in Text. *IWCS-5 Fifth International Workshop on Computational Semantics*.

Pustejovsky, James and Rob Gaizauskas (editors) (forthcoming) *Reasoning about Time and Events*. John Benjamins Publishers.

Pustejovsky, J., P. Hanks, R. Saurí, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro and M. Lazo. 2003b. The TIMEBANK Corpus. *Proceedings of Corpus Linguistics 2003*: 647-656.

Pustejovsky, J., B. Knippen, J. Littman, R. Saurí (forthcoming) Temporal and Event Information in Natural language Text. *Language Resources and Evaluation*.

Pustejovsky, James, Martha Palmer and Adam Meyers. 2005. Workshop on Frontiers in Corpus Annotation II. Pie in the Sky. ACL 2005.

Pustejovsky, J., R. Saurí, J. Castaño, D. R. Radev, R. Gaizauskas, A. Setzer, B. Sundheim and G. Katz. 2004. Representing Temporal and Event Knowledge for QA Systems. Mark T. Maybury (ed.) *New Directions in Question Answering*. MIT Press, Cambridge.

Ravichandran, Deepak and Eduard Hovy. 2002. 'Learning Surface Text Patterns for a Question Answering System'. *Proceedings of the ACL 2002*.

Saurí, Roser, Jessica Littman, Robert Knippen, Rob Gaizauskas, Andrea Setzer and James Pustejovsky. 2004. *TimeML Annotation Guidelines*. http://www.timeml.org.

Saurí, Roser and Marc Verhagen. 2005. Temporal Information in Intensional Contexts. Bunt, H., J. Geertzen and E. Thijse (eds.) *Proceedings of the Sixth International Workshop on Computational Semantics*. Tilburg, Tilburg University: 404-406.

Small, Sharon, Liu Ting, Nobuyuki Shimuzu and Tomek Strzalkowski. 2003. HITIQA, An interactive question answering system: A preliminary report. *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering.*

Soricut, Radu and Eric Brill. 2004. Automatic Question Answering: Beyond the Factoid. *HLT-NAACL 2004, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*: 57-64.

Soubbotin, Martin M. and Sergei M. Soubbotin. 2002. 'Use of Patterns for Detection of Answer Strings: A Systematic Approach'. *Proceedings of TREC-11*.

Verhagen, Marc. 2004. *Times Between the Lines*. Ph.D. thesis. Brandeis University. Waltham, MA, USA.

Verhagen, Marc and Robert Knippen. Forthcoming. TANGO: A Graphical Annotation Environment for Ordering Relations. James Pustejovsky and Robert Gaizauskas (eds.) *Time and Event Recognition in Natural Language*. John Benjamin Publications.

Verhagen, Marc, Inderjeet Mani, Roser Saurí, Robert Knippen, Jess Littman and James Pustejovsky. 2005. 'Automating Temporal Annotation with TARSQI'. Demo Session. *Proceedings of the ACL 2005*.

Voorhees, Ellen M. 2002. 'Overview of the TREC 2002 Question Answering Track'. *Proceedings of the Eleventh Text REtrieval Conference, TREC 2002*.

Voorhees, Ellen M. 2003. 'Overview of the TREC 2003 Question Answering Track'. *Proceedings of 2003 Text REtrieval Conference, TREC 2003*.