Annotating Event Implicatures for Textual Inference Tasks

Seohyun Im
Computer Science Department
Brandeis University
Waltham, MA 02453
ish97@cs.brandeis.edu

James Pustejovsky
Computer Science Department
Brandeis University
Waltham, MA 02453
jamesp@cs.brandeis.edu

Abstract

As demonstrated in recent RTE tasks and QA-based competitions (TREC), the recognition of linguistic implicatures is a critical component in any robust NLP application. There are, however, few available resources for recognizing event structure-related entailments in text. In this paper, we present a procedure for the semi-automatic construction of an Event Structure Lexicon (ESL) that can be used as a lexical resource for such tasks. The ESL is used as a resource for a subevent markup algorithm, called SUBEVITA, which creates an event implicature-annotated corpus when embedded within the TimeML-based TARSQI Toolkit. Such a resource can be used independently within the RTE task and other linguistic reasoning applications.

1 Introduction

As is well-known from the recent RTE challenge and QA-based competition (TREC), the recognition of linguistic implicatures is a critical component in complete understanding of a text for IE, QA, coreference resolution, and other NLP applications. In particular, event-based implicatures play an important role in recognizing textual inferences. Event implicature here is defined as the lexical entailment or presupposition based on the Event Structure of event-denoting expressions (e.g., verb, adjective, event nominal, etc.), consisting of pre-state, process, and result state (post-state) of an event[1].

[1]We assume the conventional formal distinctions between entailment and presupposition relations relative to an utterance. Furthermore, the Event Structure is the structure composed of lexically decomposed subevents of a matrix event denoted by an event-denoting expression: Cf. (Pustejovsky, 1995; Pustejovsky, 2000) and Moens and Steedman (1988) for a linguistic study of Event Structure.

Consider, for example, the two expressions, *the man who was killed* and *the dead man* as used in the following Text-Hypothesis pair (id 837) from the RTE1 test set.

(1) **Text:** *The Clark County medical examiner's office said* the man who was killed *was 33 years old.*
Hypothesis: *The Clark County medical examiner's office put* the dead man*'s age at 33.*

Understanding the semantic relationship between this pair of expressions requires the recognition of the entailment between *kill* and *dead*. The verb *kill* has several entailments in the following sentence:

(2) *Oswald* **killed** *Kennedy November 22, 1963.*
a. *Kennedy* died *November 22, 1963.*
b. *Kennedy* was dead **after** *November 22, 1963.*
c. *Kennedy* was alive **before** *November 22, 1963.*

All event implicatures in (a-c) above are related to the lexically encoded event structure of *kill*. The *killing* causes *dying* (see table 1); *be_dead* in (b) is a result state (post-state) of the event; and the state *be_alive* in (c) is a pre-state of the *killing* event being carried out.

In order to support the event implicature-based inferencing mentioned above, we will outline the specification and construction of a lexical resource called the Event Structure Lexicon (ESL), which encodes subevent predicate information for verbs. In the example above, such a resource would be used to identify the event *kill* as having three subevents associated with different phases of the event, where the "changes in Kennedy's state" are made explicit in the resulting annotation. We view this as an additional markup on top of TimeML (Pustejovsky et al., 2003b) procedures used for event recognition as carried out by Evita (Saurí et al., 2005). We refer to this program as SUBEVITA (**SubEv**ents **I**n **T**ext **A**nalyzer).

This paper focuses mainly on the procedure for automating the construction of such a resource, what we call an Event Structure Lexicon, ESL. The ESL is a library of context-dependent event structures for verbs consisting of: an event type; a list of subevents; a verb class specification; a subcategorization frame; and specification of semantic roles for arguments.

The automatic construction of such a resource involves the following steps: (1) Identify the verb's event type in context (its Aktionsart); (2) Assign the appropriate event structure frame associated with this event type; (3) Collect paraphrases of the predicates associated with each subevent; (4) Assemble resulting information as a structured object for each verb.

In the sections that follow, we will first briefly demonstrate the types of event implicature which can be captured by subevent encoding associated with verbs. In Section 3, we discuss how ESL can be used to support textual entailment and we compare this approach to existing lexical resources such as WordNet, VerbNet, and FrameNet. Section 4 describes the processing of constructing an ESL entry. Finally, in section 5, we describe the application of this resource in the context of an automatic annotation algorithm, SUBEVITA.

2 Reasoning from Subevents in Text

With the use of explicit temporal and event-based annotation such as that provided by TimeML (Pustejovsky et al., 2003a), it is fairly easy to see how temporal parsing can contribute to the performance of reasoning and question answering systems. Consider, for example, the textual fragment below:

(3) *Lawrence Insurance (*LI*) said it* **acquired**$_{e1}$ *United Reinsurance (*UR*) from (*USAT*) for \$28 million in March, 1989.*

A temporal parsing system allows one to answer the question of when the acquisition of UR from USAT took place, since temporal anchoring of the event is explicitly marked. The event recognition system, Evita, for example, marks *acquired* as an EVENT and the temporal parser BLINKER identifies its temporal anchoring through a TLINK with the temporal expression in the sentence, *March, 1989*.

Questions involving any reasoning over the entailments of this sentence, however, are not supported by TimeML and its associated parsers. Without identification of the lexically entailed subevents, no QA system can answer questions referring to situations before or after the event. As it happens, the *acquiring* event implies the following situations:

(4) a. *LI* **did not own** *UR before March, 1989.*
b. *USAT* **owned** *UR before March, 1989.*
c. *LI* **owns** *UR starting in March, 1989.*
d. *USAT* **does not own** *UR starting in March, 1989.*

The above sentences represent the subevents of an *acquiring* event: pre-state (4a,b) and post-state (4c,d). The subevents are themselves temporally ordered (pre-state BEFORE post-state) and are anchored in more specific relations to the temporal expression in the sentence.

The representation of subevent structure and event implicatures as meta-data markup should have a direct impact on RTE and QA system performance. Consider the queries below:

(5) a. Q: Who owned UR **before** March 1989? A: USAT
b. Q: Who owned UR **after** March 1989? A: LI

The entailed subevents are anchored in a timeline by temporal ordering between the subevents and time.

As with other change-of-state verbs, the representation of subevents and their predicates is also crucial for reasoning about the effects of movement verbs. Consider the text fragment below.

(6) a. **After** *John* **arrived** *at the hotel in the morning,*
b. *he* **left** *for climbing.*

Focusing on the verbs *arrive* and *leave*, we can assume that the subevents relevant for event-based inference include classic "ramification frames" such as those shown below:

(7) ARRIVE:
se1: pre-state: not_be_at(John, the_hotel)
se2: process: arriving(John, the_hotel)
se3: post-state: be_at(John, the_hotel)

(8) LEAVE:
se1: pre-state: be_at(John, *source*)
se2: process: leaving(John, *source*)
se3: post-state: not_be_at(John, *source*)

From the entailed subevents associated with the two path motion predicates in the text above, it is inferrable that: *John was not at the hotel, moved to the hotel, was at the hotel, left the hotel,* and finally, *is not at the hotel*. Closure over the events and temporal orderings essentially allows one to infer a general path of *John's movement* from reasoning with the subevent annotation.

Finally, similar remarks hold for inferences made due to event structure-related entailments in change-possession events such as *buy* in (9); for example, identifying the antecedent to the NP *the owner* in (10).

(9) BUY:
se1: pre-state: not_[own, possess, have](John, a_house)
se2: process: buying(John, a_house)
se3: post-state: [own, possess, have](John, a_house)

(10) a. *John* **bought** *a house last year.*
b. **The owner** *said that he will sell it next month.*

3 Previous Work

The results of the recent RTE task demonstrate that the amount of lexical and background knowledge a system is able to exploit is one of the most significant factors in the performance of a deep entailment system (Bar-Haim et al., 2006; Giampiccolo and Magnini, 2007). Further, as demonstrated above, deep knowledge of the implicatures associated with events in a text can be of great use in reasoning tasks. But, without the effort of manually encoding such a lexicon, such a resource must be created from existing databases or machine learning processing over large corpora.

Many RTE systems rely on WordNet, FrameNet, and VerbNet (Burchardt et al., 2008; Pazienza et al., 2006). WordNet (Miller, 1995) allows for some limited inferencing capabilities, but is limited by a lack of any reference to subevents or subpredicates of events. Some groups have used FrameNet (Biker et al., 1998) for the RTE task, but the result is not significantly better than simple lexical overlap (Burchardt and Pennacchiotti, 2008). One of the possible reasons is the dearth of knowledge about event entailment (e.g. *kill* $\longrightarrow$ *die*).

VerbNet (Kipper-Schuler, 2005), a hierarchical verb lexicon based on Levin's classes, has gathered recent attention within the reasoning task community. Verbs of each class in VerbNet share *syntactic frames, thematic roles,* and *selectional restrictions*. *Semantic predicates* are added to each class to better describe its semantic behaviors. It is, however, not complete or consistent as a representation of the event structure of a verb. In one recent study, for example, Zaenen et al. (2008) attempt to mine VerbNet to create consistent event structures for motion verbs in English. They demonstrate that only 28 out of 60 change_of_location verb classes have a semantic representation using Source or Destination (Goal) labels in the role names, even though these verbs are typically modeled linguistically as having Source and Goal. To overcome such shortcomings, Palmer et. al. (2009) suggest linking the VerbNet classes to an ontology where such inference rules can be associated with specific nodes in the hierarchy. This idea is similar to that presented here, in that ESL introduces hierarchical verb class structuring.

Still, one problem that remains is contextualized ambiguity of verb senses when modified by adjunct phrases, as with the ambiguity of motion verbs triggered by their combination with prepositions. For example, the verb *run* in the sentence *John ran fast* is an activity verb which has only a process subevent (*running(John)*). On the other hand, it changes its verb class into a change_of_location class as an accomplishment with a preposition as in *John ran into the store*. As a result, the verb *run* has 3 subevents: pre-state (*not_be_at(John, the_store)*), process (*running(John)*), and post-state (*be_at(John, the_store)*). Hence, disambiguation of motion verbs is dependent on their adjunct composition with prepositional phrases and particles.

4 Building Subevent Structures

As mentioned in the previous section, using existing lexical resources such as Wordnet or VerbNet to create subevent lexical frames for verbs is not in itself an adequate solution. In this section, we describe a semi-automated procedure for constructing a lexicon of event-based implicatures (ESL) using a combination of corpora and lexical resources. For each verb, this involves the following steps:

(11) a. Identify the "event type in context" (the contextualized Aktionsart);
b. Assign the appropriate subevent structure frame associated with this event type;
c. Collect paraphrases of the predicates associated with each subevent;
d. Assemble resulting information as a structured object for each verb into ESL.

The first task is "Event Type Identification", which is to identify the aspectual class of each verb as it occurs in context in text. It has long been acknowledged that aspectual class and event structure are important attributues for verb behavior and semantics (Vendler, 1967; Dowty, 1979; Pustejovsky, 1995), yet recognizing the aspectual class of a verb in context has proven difficult (Klavans and Chodorow, 1992). Recently, however, Zarcone and Lenci (2008) demonstrate that robust automatic event type (Aktionsart) classification is possible.

The event type of a verb occurrence is determined by the complex interaction among different features such as the verb's argument structure, its aspect, the definiteness and plurality of its arguments, frequency and genericity marking, and so on (Zarcone and Lenci, 2008). For example, the progressive aspect cancels the result state of a lexically-marked accomplishment event and thus changes its event type to a process (e.g. *build*: transition; *be_building*: process).

Following Zarcone and Lenci (2008), verb occurrences in the corpus are manually annotated with their proper event types. Then, Maximum Entropy classifiers are applied and trained on the corpus. We modify the classifier slightly, however, to match our task. First, event type is simplified to a three-way distinction of process, state, and transition, with no distinction between achievement and accomplishment since their basic event structure frames are the same: pre-state, process, and post-state.[2] Further, we do not consider modality or negation as linguistic features.

Once an event type is identified, an event structure frame (ESF) is assigned for that verb's semantic class. We assume that (i) verb occurrences are classified into verb classes, and (ii) each verb class has its own proper event structure frame. We present the context-dependent verb class and their event structure frames for the verbs WALK, BELIEVE, GIVE, and KILL in table 2.

We assume a model of event structure as presented in Generative Lexicon (GL) (Pustejovsky, 1995) and further developed in (Pustejovsky, 2000). The event structure frame in GL is a representation associated with a verb where the predicative content is decomposed into subevents and their temporal ordering, along with headedness. The basic event types are: process, state, and transition (achievement and accomplishment). A transition consists of pre-state, process, and result state (post-state). We use the information for our system.

The event structure for a verb is determined contextually in text. For example, the manner_of_motion verb *run* is lexically a process verb (e.g. *John walked quickly*). However, it changes into the change_of_location verb class in a context such as *The student walked into the store*, which has the three subevents associated with a transition.

The verb classes and subclasses are based on the Brandeis Semantic Ontology, BSO (Pustejovsky et al., 2006a). Each of the subclasses has its own event structure frame assigned. The upper level verb class consists of: process, state, change_of_location, change_of_possession, and change_of_state. All classes except for state have their corresponding causation verb classes. Each of these, in turn, may have subclasses. Specifically, the change_of_location class can be divided into from_source, to_goal, from_source_to_goal, etc., which are being developed in the broader context of modeling motion in language (Pustejovsky and Moszkowicz, 2008).

After the assignment of an event structure frame to a verb, we compile paraphrases for the predicates associated with each subevent in the event structure. That is, how are the different phases of the event expressed lexically in the language? For this step, we utilize the lexical resources of WordNet and Extended WordNet.

For paraphrasing, we distinguish two predicative classes: *closed domain* and *open domain*. Closed domain predicates are verbs falling into semantic classes with generally well-defined predications associated with the subevents. This includes, for example, the verb classes change_of_location and change_of_possession. The event structure frame for the former class is shown below, where pred is the verb assigned to the class:

(12) pred:change_of_location(x,y)
se1: pre-state: not_be_in(x,y)
se2: process: pred-ing(x,y)
se3: post-state: be_in(x,y)

[2] Some semantic distinctions between accomplishments and achievements are obviously of importance to subsequent linguistic reasoning, but both classes are subject to the "imperfective paradox", and as a result the distinction can be ignored for most inference.

Text	*kill(x,y)*	
ETC	transition	
V_Class	change_of_state	
Subclass	destruction	
ESF	se1: pre-state: not_be_pred-pp(y) se2: process: pred-ing(x,y) se3: post-state: be_pred-pp(y)	
Paraphrase ArgLink	se1: pre-state: not_be_killed, [alive, living](y) se2: process: [kill, causing to die, causing to decrease, causing to perish, causing to pass away, causing to expire, exterminating](x,y) se3: post-state: be_[died, deceased, perished, passed away, expired, dead](y)	

Table 1: Example of Event Structure Lexicon Entry

Verb	Event Type	Lexical class	Contextual class	subclass	ESF	Example
WALK	process	process	process	process	se1:pred-ing	*We walked aerobically*
WALK	transition	process	change_of_loc	to_goal	se1: not_be_at se2: pred-ing se3: be_at	*We walked to the store*
				from_source	se1: be_at se2: pred-ing se3: not_be_at	*He walked from the park*
				from_source_to_goal	se1: be_at se2: not_be_at se3: pred-ing se4: not_be_at se5: be_at	*He walked from home to school*
				two_side_path	se1: be_at_start_point se2: pred-ing se3: be_at_end_point	*John walked across the room*
BELIEVE	state	state	state	psych_state	se1: pred	*I believe you can do it*
GIVE	transition	change_of_pos	change_of_pos	transfer_possession	se1:pre-state: possess se2:pre-state: not_possess se3:process: pred-ing se4:process: being_pred-pp se5:post-state: not_possess se6:post-state: possess	*She gave me an extra pillow*
GIVE	process	change_of_pos	process	process	se1: pred-ing	*She gave a scream of delight*
KILL	transition	change_of_state	change_of_state	destruction	se1: not_be_pred-pp se2: pred-ing se3: being_pred-pp	*They killed at least 40 people*
KILL	transition	change_of_state	state	state	se1: pred	*Stress kills*

Table 2: Event Type, Verb Class, and Event Structure Frame

For example, identifying the verb *drive* as a change_of_location verb generates the closed domain ESL entry shown below:

(13) drive in *John drove to Boston*
se1: pre-state: not_be_in(x,y)
se2: process: driving(x,y)
se3: post-state: be_in(x,y)

Given this basic ESL in (13), the only paraphrases available involve synonyms for the $se2$ predicate, *driving*, such as *going* and *traveling*.

Open domain predicates include verbs in the large change_of_state class, where there are few if any general predications associated with subevents in the event structure. The event structure frame associated with such a class is shown in (14) with an example for the verb *die* in (15).[3]

(14) pred:change_of_state(x)
se1: pre-state: not_be_pred-pp(x)
se2: process: pred-ing(x)
se3: post-state: be_pred-pp(x)

(15) die in *The plants died.*
se1: pre-state: not_have_died(x)
se2: process: dying(x)
se3: post-state: have_died(x)

After a verb is identified with a particular open domain verb class, paraphrases are generated for each subevent in the event structure frame. For example, for $se1$, not_have_died, we generate not_dead, alive, and so forth. Similarly, for $se3$, have_died, we generate not_alive and dead. Thus, all predicates of the subevents are lexically enriched through paraphrasing with the help of various resources such as WordNet.

[3]As pointed out by one reviewer, it might be possible to further define the subclasses of open domain predicates, thereby allowing them to have closed domain behavior. That is, one could distinguish between degree-change, creation, and destruction predicates more specifically. We are currently pursuing this possibility.

The last step involves compiling the extracted event structure frames of verb occurrences into the ESL. Table 3 shows the ESL for the verb *acquire* in (8), compared with the semantic frame of *obtain* in VerbNet[4]. As we see in the next section, SUBEVITA uses the ESL as a lexical resource for markup of SUBEVENT tags.

5 Annotating Text with ESL

Using the ESL as a reference library, a subevent annotation algorithm called SUBEVITA is now able to annotate an EVENT-tagged corpus such as TimeBank with SUBEVENT tags to represent the event structure frames of EVENT-tagged expressions. SUBEVITA takes text that has been processed by a temporal parsing systems such as TTK (Verhagen and Pustejovsky, 2008), with EVENT and TIMEX3 tags explicitly annotated, and generates the appropriate subevent tags for each event. We can think of SUBEVENT tagging as a general, domain-independent meta-data enrichment of text, which can be exploited by diverse NLP applications, such as RTE, QA, and other such tasks. We will not elaborate on SUBEVITA here, but the output of this process is illustrated below, with a text fragment containing the verb *acquire*.

(16) *LI* **said**$_{e1}$ *it*$_{a1}$ **acquired**$_{e2}$ *UR*$_{a2}$ *from USAT*$_{a3}$ *for $28 million in* **March, 1989**$_{t1}$.

< EVENT eid="e2" class="OCCURRENCE" tense="PAST" aspect="NONE" polarity="POS"> **acquired** </EVENT>
<TIMEX3 tid="t1" type="DATE" value="1989-03"> **March, 1989** </TIMEX3>

Evita annotates *acquired* with an EVENT tag and assigns the appropriate attribute-value pairs. According to its ESL entry, the verb *acquire* has five subevents and thus SUBEVITA inserts five SUBEVENT tags as meta-data markup, based on the ESL in Table 3.

<SUBEVENT seid="se1" partOf="e2" />
<SUBEVENT seid="se2" partOf="e2" />
<SUBEVENT seid="se3" partOf="e2" />
<SUBEVENT seid="se4" partOf="e2" />
<SUBEVENT seid="se5" partOf="e2" />

SUBEVITA connects the appropriate arguments of the verb in text with SUBEVENTs via ARGLINK tags (Pustejovsky et al., 2006b). The resulting meta-data annotation of this text now enables the inferencing capabilities mentioned in Sections 1 and 2 above. That is, entailments and questions referring to the subevent implicatures of the transaction can now be addressed, by virtue of the explicit representation of these events in the annotation through the ESL.

[4]VerbNet shows the semantic frame of *obtain* for the verb class which *acquire* belongs to.

6 Conclusion

In this paper, we presented a procedure for semi-automating the construction of an Event Structure Lexicon (ESL) that can be used as a lexical resource for inference-related tasks in NLP. The ESL is used as a resource for a subevent markup algorithm, called SUBEVITA, which creates an event implicature-annotated corpus when embedded within the TimeML-based TARSQI Toolkit. Such a resource can be used independently within the RTE task and other language reasoning applications. The present work is obviously programmatic and is still in development. Some of the risks and uncertainties in the above technique include: overgeneration of paraphrases for each subevent predicate; misclassification of the event type in context; and misclassification of the verb class, due to lexical ambiguity. These are matters we hope to address in the near future.

Acknowledgement

This work was supported in part by grants to J. Pustejovsky by the NGA and MITRE. We would like to thank Anna Rumshisky, Jessica Moszkowicz, Roser Saurí, and Marc Verhagen.

References

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Collin F. Biker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of COLING-ACL*, Montreal, Canada.

Aljoscha Burchardt and Marco Pennacchiotti. 2008. Fate: A framenet-annotated corpus for textual entailment. In *Proceedings of LREC 2008*, Marrakech, Morocco.

A. Burchardt, M. Pennacchiiotti, S. Thater, and M. Pinkal. 2008. Assessing the impact of frame semantics on textual entailment. *Journal of Natural Language Engineering*. Special Issue on Textual Entailment.

ESL		VerbNet	
VERB	**ACQUIRE**	VERB	**OBTAIN**
CLASS	change_of_possession	CLASS	obtain-13.5.2
SUB_CLASS	get_possession	SUB_CLASS	
EVENT_TYPE	transition		
SUB_CAT	NP V NP from_NP		NP v NP PP.source
SEM_ROLE	agent theme source	SYNTAX	agent V theme {from} source
SUBEVENT	se1: pre-state: not_[possess, have, own] (LI, URR) se2: pre-state: [possess, have, own] (USAT, URR) se3: process: [acquiring, getting, gaining] (LI, URR, USAT) se4: post-state: [possess, have, own] (LI, URR) se5: post-state: not_[possess, have, own] (USAT, URR)	SEMANTICS	has_possession(start(e), ?source, theme) transfer(during(e), theme has_possession(end(e), agent, theme) cause(agent, e)
TEMP_ORDER	se3 ENDS se1 se3 BEGINS se4 se3 ENDS se2 se3 BEGINS se5		
SENTENCE	*LI acquired URR from USAT for $28million in March, 1989.*	EXAMPLE	*Carmen obtained the spare part from Diana.*

Table 3: ESL of *acquire* in (8) vs. *obtain* in VerbNet

David Dowty. 1979. *Word Meaning and Montague Grammar*. Reidel, Dordrecht.

Danilo Giampiccolo and Bernardo Magnini. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June. Association for Computational Linguistics.

Karin Kipper-Schuler. 2005. *VerbNet: A broad coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania, USA.

Judith L. Klavans and Martin Chodorow. 1992. Degrees of stativity: The lexical representation of verb aspect. In *COLING 1992*, pages 1126–1131.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, November.

Marc Moens and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15–28.

Martha Palmer, Jena D. Hwang, Susan Windisch Brown, and Karin Jasmine Zhou. 2009. Leveraging lexical resources for the detection of event relations. In *Proceedings of AAAI 2009 Spring Symposium: Learning by Reading and Learning to Read*, Colorado, USA, March. AAAI.

Maria Teresa Pazienza, Marco Pennacchiotti, and Fabio Massimo Zanzotto. 2006. Mixing wordnet, verbnet and propbank for studying verb relations. In *Proceedings of the 5th Language Resource and Evaluation Conference*, Genova, Italy.

James Pustejovsky and Jessica L. Moszkowicz. 2008. Integrating motion predicate classes with spatial and temporal annotations. In *COLING 2008: Companion volume - Posters and Demonstrations*, pages 95–98, Manchester, August.

James Pustejovsky, Jose Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003a. Timeml: Robust specification of event and temporal expressions in text. In *Proceedings of IWCS-5, Fifth International Workshop on Computational Semantics*.

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

James Pustejovsky, Catherine Havasi, Jessica Littman, Anna Rumshisky, and Marc Verhagen. 2006a. Towards a generative lexical resource: The brandeis semantic ontology. In *Proceedings LREC 2006*, Genoa, Italy.

James Pustejovsky, Jessica Littman, and Roser Saurí. 2006b. Argument structure in timeml. In *Dagstuhl Seminar Proceedings: Annotating, Extracting, and Reasoning about Time and Events*.

James Pustejovsky. 1995. *The Generative Lexicon*. The MIT Press., USA.

James Pustejovsky. 2000. Events and the semantics of opposition. In James Pustejovsky and Carol Tenny, editors, *Events as Grammatical Objects*. CSLI Publciations.

Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovksy. 2005. Evita: A robust event recognizer for qa systems. In *Proceedings of the HLT/EMNLP 2005*, pages 700–707.

Zeno Vendler. 1967. *Linguistics in Philosophy*. Cornell University Press, Ithaca, NY.

Marc Verhagen and James Pustejovsky. 2008. Temporal processing with the tarsqi toolkit. In *In Proceedings of Coling 2008: Companion Volume-Posters and Demonstractions*, pages 189–192.

Annie Zaenen, Cleo Condoravdi, and Danny Bobrow. 2008. The encoding of lexical implications in verbnet. In *Proceedings of LREC 2008*, Morocco.

Alessandra Zarcone and Alessandro Lenci. 2008. Computational models of event type classification in context. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA).