Annotating Event Implicatures for Textual Inference Tasks

Seohyun Im and James Pustejovsky

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

Introduction

- Event Implicature
  - The lexical entailment or presupposition based on the Event Structure of event-denoting expressions
  - A critical component in complete understanding of a text for NLP applications (QA, IE, Coreference Resolution, etc.)

- Event Structure (GL-based)
  - The structure composed of lexically decomposed subevents of a matrix event denoted by an event-denoting expression (Pustejovsky, 1995; 2000)
  - consists of pre-state, process, and result state (post-state)

Text-Hypothesis pair (RTE 1 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.*

(2) *Oswald* **Killed** *Kennedy November 22, 1963*.
a. *Kennedy* died *November 22, 1963*. (caused)
b. *Kennedy* was dead ***after*** *November 22, 1963.* (post-state)
c. *Kennedy* was alive ***before*** *November 22, 1963.*(pre-state)

Our Work

- To support the event implicature-based inferencing
- **Event Structure Lexicon** (ESL)

  : A lexical resource which encodes subevent predicate information for verbs.

  For example, the ESL is used to identify the event ***kill*** as having 3 subevents associated with different phases of the event, where the "changes in Kennedy's state" are made explicit in the resulting annotation.
- **SUBEVITA** (SubEvents In Text Analyzer)

  : Additional SUBEVENT markup on top of TimeML (Pustejovsky et. al., 2003) procedures used for event recognition as carried out by Evita (Saurí et. al. 2005).
- This paper focuses mainly on the procedure for automating the construction of the ESL.

Event Structure Lexicon (ESL)

- A library of context-dependent event structures for verbs.
- It consists of:
  - an event type
  - a list of subevents
  - a verb class specification
  - a subcategorization frame
  - specification of semantic roles for arguments
  - temporal order between subevents
  - example sentence

Event Structure Lexicon (ESL)

- **Automatic Construction**
  - Identify the verb's event type in context (Aktionsart)
  - Assign the appropriate event structure frame associated with this event type
  - Collect paraphrases of the predicates associated with each subevent
  - Assemble resulting information as a structured object for each verb

Outline

- Reasoning from Subevents in Text
- Previous Work
- Building Subevent Structures (ESL)
- Annotating Text with ESL (SUBEVITA)
- Event Type Identification experiment for Motion verbs
- Conclusion

Types of Event Implicature (1):
Temporal relation between subevents and time

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

- A temporal parsing system (e.g. TTK) allows one to answer the question of when the acquisition of UR from USAT took place.
- TimeML and TTK
  - Evita marks *acquire* as an EVENT
  - BLINKER identifies its temporal anchoring through a TLINK with *March, 1989*.
  - But TimeML and TTK canNOT support the entailment-related questions.

Cont.

(4) Subevents of *acquiring*

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.*

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

b. Q: Who owned UR after March 1989? A: LI

On the assumption that the subevents in (4) are themselves temporally ordered and are anchored in temporal relations to *March 1989*, QA system can answer to the questions in (5)

Therefore, event implicature is important for temporal annotating and reasoning systems such as TimeML and TTK.

Types of Event Implicature (2): Classic "ramification frames"

(6) a. **After** *John* **arrived** *at the hotel in the morning*.

b. *he* **left** *for climbing*.

(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*)

Cont.

(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*)

- **Path of *John's movement***

  John was not at the hotel -> moved to the hotel -> was at the hotel -> left the hotel -> finally, is not at the hotel

Types of Event Implicature (3): *change_possession* events

(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.*

Previous Work

- The result of the recent RTE task
  - The amount of lexical and background knowledge is important for the performance of a deep entailment system.

    (Bar-Haim et. al., 2006; Giampiccolo and Magnini, 2007)
  - Event implicature in a text can be of great use in reasoning tasks
- But…

  Such a resource should be constructed automatically from existing database or machine learning processing over large corpora.

Previous Work

- **WordNet** (Miller, 1995)
  - allows for some limited inferencing capabilities.
  - but is limited by a lack of any reference to subevents.
- **FrameNet** (Biker et. al., 1998)
  - The result of the RTE task using FrameNet is not significantly better than simple lexical overlap (Burchardt and Pennacchiotti, 2008)
  - The dearth of knowledge about event entailment (e.g. *kill* → *die*)

Previous Work

- **VerbNet** (Kipper-Schuler, 2005)
  - A hierarchical verb lexicon based on Levin's classes
  - Not complete or consistent as a representation of the event structure of a verb.
  - Zaenen et al. (2008)
    - attempt to mine VerbNet to create consistent event structures for motion verbs in English
    - only 28 out of 60 change_of_location verb classes have a semantic representation using Source or Destination (Goal) labels.
    - To overcome such shortcomings, Palmer et al. (2009) suggest linking the VerbNet classes to OntoNote.

Previous Work

- Contextualized ambiguity of verb senses
  - *John* ***ran*** *fast.*
    - event type: process
    - verb class: activity
  - *John* ***ran into*** *the store.*
    - event type: accomplishment
    - verb class: change_of_location

Therefore, disambiguation of motion verbs is dependent on their adjunct composition with prepositional phrases and particles.

Event Structure Lexicon (ESL)

- **Automatic Construction**
  - Identify the verb's event type in context (the contextualized Aktionsart)
  - Assign the appropriate subevent structure frame associated with this event type
  - Collect paraphrases of the predicates associated with each subevent
  - Assemble resulting information as a structured object for each verb

Event Type Identification

- To identify the aspectual class of each verb as it occurs in context in text
- Zarcone and Lenci (2008)
  - Automatic recognition of the contextually-determined event type of a verb occurrence in text, using MaxEnt model
  - annotated the Italian corpus manually
  - Maximum Entropy classifiers are applied and trained on the corpus.
  - Event Types: state, process, achievement, accomplishment
  - Features: tense, aspect, adverbial (temporal, frequency, iterative, intention), presence of arguments (or subj, obj, locative, complemental sentence), passive diathesis, number, animacy, definiteness

Event Type Identification

- We adopt Zarcone and Lenci (2008)
- Changes
  - Event Type: process, state, transition (No distinction between achievement and accomplishment.)
  - use the result of parsing with Stanford dependency parser (POS, PennTree, dependency)
  - Features: POS, aspect, voice (passive auxiliary), kinds of prepositions, part of semantic role information (locative, direction, path, time), presence of direct object and indirect object, animacy (people, object, abstract), definiteness, number, plurality, frequency
  - We do NOT consider modality or negation as features.

Assignment of the Event Structure Frame

- Assumption
  - Verb occurrences are classified into verb classes.
  - Each verb class has its own proper event structure frame.
- Based on GL (Pustejovsky, 1995;2000)
- Event Structure: pre-state, process, post-state
- Contextually-determined

Event Type, Verb Class, and Event Structure Frame Example

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

Cont.

- Verb Class and Subclasses are based on the Brandeis Semantic Ontology (Pustejovsky et al., 2006)
- BSO-based Verb Class

  process, state, change_of_location, change_of_possession, change_of_state

  - Each class has subclasses
  - For example, **change_of_location** class can be divided into **from_source, to_goal, from_source_to_goal**, etc.
  - being developed in the broader context of modeling motion in language (Pustejovsky and Moszkovicz, 2008)
  - All classes except for **state** have their corresponding causation classes

Paraphrasing

- After the assignment of an event structure frame to a verb, we compile paraphrases for the predicates of each subevent in the event structure.
- All predicates of the subevents are lexically enriched through paraphrasing with the help of various resources such as WordNet.

Event Structure Lexicon Entry

Text	**Kill** (x,y)
Event Type	transition
Verb class	Change_of_state
Subclass	destruction
Event Structure	Se1: pre-state: **not_be_pred-pp** (y) Se2: process: **pred-ing** (x, y) Se3: post-state: **be_pred-pp** (y)
Paraphrasing	Se1: pre-state: **not_be_killed,** [alive, living] (y) Se2: process: **killing,** [causing to die, causing to decease, causing to perish, causing to pass away, causing to expire, exterminating] (x, y) Se3: post-state: **be_killed,** be_[died, deceased, perished, passed away, expired, dead] (y)

Paraphrasing

Closed domain vs. Open domain

- For paraphrasing, we distinguish two predicative classes: closed domain and open domain

- **Closed domain**
  - closed domain predicates are verbs falling into semantic classes with generally well-defined predications associated with the subevents
  - Example

    Change_of_location

    Change_of_possession verb classes

Closed domain

(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)

(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)

Open domain

- **open domain**
  - There are few if any general predications associated with subevents in the event structure.
  - need to paraphrase predicates of subevents.
  - Example: *change_of_state* verb class

Open domain

(14) pred: change_of_state

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)

* It might be possible to further define the subclasses of open domain predicates, thereby allowing them to have closed domain behavior.

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

ESL		**VerbNet**	
VERB	**ACQUIRE**	VERB	**OBTAIN**
CLASS	Change_of_possession	CLASS	Obtain-13.5.2
SUBCLASS	Get_possession	SUBCLASS	
E_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 $2 million in March, 1989.*	EXAMPLE	*Carmen obtained the spare part from Diana.*

SUBEVITA

- Using the ESL as a reference library, SUBEVITA annotates an EVENT-tagged corpus (e.g., TimeBank) with SUBEVENT tags to represent the event structure frames of EVENT-tagged expressions.
- It takes the text that has been processed by a temporal parsing system such as TTK. (Verhagen and Pustejovsky, 2008)
- SUBEVENT tagging: A general, domain-independent meta-data enrichment of text

SUBEVENT tagging

(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>

<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" />

Event Type Identification Experiment for motion verbs

- Corpus
  - About 30K words, 100 documents from Traveler’s blog
  - used about 15K words, 50 documents for training
- Preprocessing
  - parsing with Stanford Dependency Parser
  - POS, PennTree, Dependency
- Annotated the event type of each motion verb occurrence from the parsed corpus manually.
- MaxEnt Classifier: Ben Wellner’s Carafe

Motion Verbs from the Corpus

- List
  - appear(3) arrive(39) board(3) bring(9) carry(2) catch(8) climb(4) come(31) continue(3) cross(6) cycle(2) depart(2) dive(2) drive(17) drop(5) enter(4) escape(2) explore(7) fall(5) float(2) fly(9) get(75) go(76) hang(2) head(32) hop(2) jump(4) land(2) leave(26) load(2) move(9) park(2) pass(9) pedal(2) pour(2) proceed(2) progress(2) pull(2) put(3) reach(4) return(9) ride(2) roll(2) run(6) show(3) sit(9) stand(8) travel(12) walk(17) wander(6)
  - accompany(1) approach(1) ascend(1) chuck(1) creep(1) dance(1)descend(1) drag(1) draw(1) dump(1) eject(1) emerge(1) hand(1) jolt(1) lead(1) lie(1) march(1) migrate(1) moor(1) originate(1) part(1) persevere(1) rise(1)rush(1) sail(1) seat(1) set(1) settle(1) skip(1) soar(1)steer(1) step(1) stretch(1)surf(1) swing(1)throw(1) toss(1)trot(1) tumble(1)

The composition of the corpus

Verbs	TOTAL	STATE	PROCESS	TRANSITION
90	654	20	168	466

Feature Set

- based on the result of dependency parsing
- Features
  - POS, lemma, aspect, preposition
  - the semantic property of the argument of preposition (locative, time, direction, path)
  - auxpass (passive voice)
  - the semantic property of direct object (locative, time, people, abstract, object), indirent object
  - the semantic property of subject (locative, time, path, people, abstract, object)
  - presence of complement (xcomp, ccomp)
  - locative adverbial
  - Number

The Result

- excluded the verbs with only one occurrence for test.

	Precision	Recall	F-measure
STATE	0.70	0.69	0.72
PROCESS	0.77	0.73	0.75
TRANSITION	0.83	0.86	0.85

Conclusion

- We presented a procedure for semi-automating the construction of an Event Structure Lexicon (ESL)
- ESL can be used as a lexical resource for inference-related tasks in NLP.
- ESL is used as a resource for a subevent markup algorithm (SUBEVITA).
- The present work is still in development.
- The risks
  - Overgeneration of paraphrases
  - Misclassification of the event type in context
  - Misclassification of the verb class, due to lexical ambiguity

Acknowledgement

- supported in part by grants to J. Pustejovsky by the NGA and MITRE
- Thanks to Anna Rumshisky, Jessica Moszkowicz, Roser Saurí, and Marc Verhagen.