Integrating Generative Lexicon Event Structures into VerbNet

Susan Windisch Brown[1], James Pustejovsky[2], Annie Zaenen[3], Martha Palmer[1]
[1]University of Colorado, [2]Brandeis University, [3]Stanford University
{susan.brown, martha.palmer}@colorado.edu, jamesp@cs.brandeis.edu, azaenen@stanford.edu

Abstract

This paper focuses on specific changes to the semantic representations associated with classes of verbs in the English lexical resource VerbNet (Schuler, 2005). The new form has been restricted to first-order representations to simplify use by and integration with planners. More significantly, the modifications incorporate the Generative Lexicon's event structure, with temporal ordering of subevents associated with explicit predications over the verb's arguments. These changes allow for greater flexibility in representing complex events, for a more consistent treatment of the oppositions inherent in change-of-state classes, and for a more nuanced portrayal of the Agent's role.

Keywords: semantics, event representation, lexicon

1. Introduction

Natural language processing has been moving from shallow semantic parsing to deeper semantic analysis of text in order to better support knowledge representation and reasoning systems. Understanding sentences requires more than identifying events and participants and giving them thematic role labels. In particular, it is essential to recognize any temporal sequencing within the event and any changes in state that might have occurred (Pustejovsky, 2005; Mani and Pustejovsky, 2012).

The language resource VerbNet (Kipper et al., 2006) is a promising source of such information. It is a hierarchical, domain-independent verb lexicon that groups verbs into classes based on similarities in their syntactic and semantic behavior (Schuler, 2005). Each class in VerbNet defines a set of members, thematic roles for the predicate-argument structure of these members, selectional restrictions on the arguments, and frames consisting of a syntactic description and a corresponding semantic representation. It has long been used for semantic role labeling and other inference-enabling tasks (Shi and Mihalcea, 2005; Giuglea and Moschitti, 2006; Loper et al., 2007). Automatic disambiguation of a verb's VerbNet class has also improved (Abend et al., 2008; Brown et al., 2014; Kawahara and Palmer, 2014). Efforts to use its semantic representations (Zaenen et al., 2008; Narayan-Chen et al., 2017) have revealed a need to revise them for consistency and greater expressiveness, in particular, a clearer representation of subevents.

Recent work in Generative Lexicon (GL) has focused on further articulating the semantics of subevent structure in language (Pustejovsky and Moszkowicz, 2011; Pustejovsky, 2013). Hence a reasonable undertaking is to revise VerbNet to take advantage of GL's progress in representing subevent structure while preserving VerbNet's strengths in linking predicate argument structure, thematic roles and semantic representations.

The remainder of this paper will describe the changes being made to VerbNet's semantic representations and the reasons behind those changes. Section 2 briefly describes the role semantic representations play in VerbNet and breaks down the structure of the "old" semantic representations in version 3.3. Section 3 goes into more detail on the drawbacks of these representations and the requests from users for new functionality. An overview of GL event structure is given in section 4, highlighting how it can fulfill the needs identified in section 3. Section 5 uses VerbNet change of location and change of state classes to illustrate the new representations to be released in VerbNet version 4.0.

2. VerbNet and Its Representation of Events

Each VerbNet class contains semantic representations that are compatible with the member verbs and the syntactic frames of the class. This pairing of each syntactic frame in a class with a semantic representation is a unique feature of VerbNet that emphasizes the close interplay of syntax and semantics. The semantic information is expressed as a conjunction of semantic predicates, such as **motion, perceive** or **cause** and an event variable **E**. Some of these are meant to describe the participants during various stages of the event evoked by the syntactic frame. For example, one of the intransitive frames in the class Run-51.3.2 is shown in (1), with the final 4 lines making up the semantic representation:

(1) *The horse ran into the barn.*
NP V PP
Theme V Destination

motion(during(E), Theme)
path_rel(start(E), Theme, Initial_location, ch_of_loc, prep)
path_rel(during(E), Theme, Trajectory, ch_of_loc, prep)
path_rel(end(E), Theme, Destination, ch_of_loc, prep)

The arguments of each predicate are represented using the thematic roles for the class. These roles provide the link between the syntax and the semantic representation. Each participant mentioned in the syntax, as well as necessary but unmentioned participants, is accounted for in the semantics. For example, the second component of the first

path_rel semantic predicate above includes an unidentified Initial_location.[1] Temporal sequencing is indicated with the second-order predicates start, during and end, which are included as arguments of the appropriate first-order predicates. A similar sentence with an Agent causing the motion, such as *John herded the sheep into the barn,* would add **cause**(Agent, E) to the semantic representation in (1).

The semantic representations associated with a class capture generalizations about the semantic behavior of the member verbs as a group. For some classes, such as the Battle-36.4 class, the verbs are semantically quite coherent (e.g., *battle, skirmish, war*) and the semantic representation is correspondingly precise.

(2) *Sparta warred with Athens.*
NP V PP
Agent V {with} Co-Agent

social_interaction(during(E), Agent, Co-Agent)
conflict(during(E), Agent, Co-Agent)
possible_contact(during(E), Agent, Co-Agent)
manner(Hostile, Agent, Co-Agent)

Other classes, such as Other Change of State-45.4, contain widely diverse member verbs (e.g., *dry, gentrify, renew, whiten*). The representation must be very general to apply to all the verbs. The following representation for this class ignores the specific type of state change in the example sentence (i.e., from wet to dry) in order to be general enough for any verb in the class when used in a basic transitive sentence.

(3) *John dried the clothes.*
NP V NP
Agent V Patient

path_rel(start(E), Initial state, Patient, ch_of_state, prep)
path_rel(result(E), Result, Patient, ch_of_state, prep)
cause(Agent, E)

Additional, more precise semantic information can be associated with each individual verb in a class. To that end, verb-specific features that differentiate member verbs within a class are currently being added to certain classes (Pustejovsky et al., 2016). Our goal in the revisions described here, however, is to enhance the accuracy and expressiveness of a class's representations while preserving the generalizations that can be made across all the verbs in a class.

3. The Impetus for Change

Over the years, VerbNet has undergone several revisions, either to expand its coverage (Kipper et al., 2008), to improve the clarity and consistency of its components (Bonial et al., 2011; Hwang, 2014), or in response to users' needs. Minor changes had been made to VerbNet's semantic representations, such as adding new semantic predicates, consolidating similar predicates, and standardizing the types and number of arguments particular predicates take. Efforts to use the semantic predicates in various tasks revealed several weaknesses that led us to undertake this current revision.

Zaenen et al. (2008) used VerbNet predicates to inform change of location inferences, such as inferring from the statement *The diplomat left Bhagdad* that before the event, the diplomat was in Bhagdad. This is exactly the sort of information VerbNet's semantic representations were designed to provide. However, such information was not consistently supplied across all of the classes that dealt with motion. For several motion classes, End(E) was given but not Start(E). Other classes that dealt with the change of location of participants relative to each other (e.g., gather, mix) did not include a motion predicate at all. Although they found that in many cases VerbNet did support adequate inferencing, its inconsistencies lessened its usefulness.

Several of the omissions that were found stemmed from the practice of only including syntactic arguments in the semantic representations, ignoring any possible adjuncts. Including more semantic information about subevents in the representation, whether that information is instantiated in the syntax or not, was an important desideratum for the new representations.

A recent effort to use VerbNet in human-computer interaction (Narayan-Chen et al., 2017) found that a few aspects of the semantic representation could be altered both to facilitate the interaction between the language parsing and the planning components of the system and to increase the expressiveness of the representation.

First, attempts to use VerbNet information in robotics applications showed the need for a switch to a first-order logic representation. In addition, more specific causal and temporal relations were desired. The current method of indicating causation, for example, simply had an Agent and the event variable E as arguments to a Cause predicate. This is somewhat misleading in that it could imply that the Agent causes all of (E), including whatever state exists at Start(E).

Second, the temporal sequencing of an event is sometimes more complex than what can be expressed with only Start(E), During(E) and End(E). For example, a 'throw' event involves a Theme in motion during the event and an Agent in contact with Theme, but only in contact for part of the period denoted by During(E). The ability to subdivide these three periods seemed in order. For some events of this nature, we attempted to show this sequence in version 3.3 by positing two events, E_0 and E_1, and temporally subdividing those with During and End:

(4) *Mary threw the ball.*
NP V NP
Agent V Theme

exert_force(during(E_0), Agent, Theme)
contact(end(E_0), Agent, Theme)
¬ **contact**(during(E_1), Agent, Theme)
motion(during(E_1), Theme)
cause(Agent, E_1)

[1] Each **path_rel** predicate also has a **prep** slot which is a placeholder for the information contributed by specific prepositions to be passed to inferencing for further semantic processing.

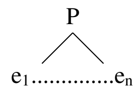
As we will see, GL event structure and temporal sequencing of subevents solves this problem more logically and transparently, while preserving the idea that this sentence describes only a single event E.
Finally, representing simultaneity of subevents or gaps in the temporal sequencing is difficult with the apparently continuous sequence of before, during and after. In a cutting event, for example, an Agent is performing an action that results in a change of state in the Patient. The end of the cutting action temporally meets the beginning of the final state of the Patient. But this is not always the case in a causal event. With *John dried the clothes,* John may be doing something, such as waving a hair drier in front of the clothes, that does continue until the final end state is reached. Or John may have hung the clothes outside on a clothes line, in which case, his action does not temporally meet the final "dry" end state. For the representation of a generic "dry" event, we do not want an assumption that the actions taking place During(E) are necessarily contiguous with End(E). Therefore, in addition to having a means of identifying more subevents, we would like to have a means of indicating the specific temporal relations between them.

4. Generative Lexicon's Event Structure

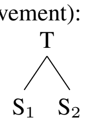
Many of the issues described in section 2 are resolved by adopting aspects of the event structure as modeled in Generative Lexicon. Classic GL characterizes the different Aktionsarten in terms of structured subevents (Pustejovsky, 1995). Different event types can be represented as typed feature structures or in the form of tree structures, as below.

(5) a. STATE: a simple event, evaluated without referring to other events: *be sick*, *love*, *know*

b. PROCESS: a sequence of events identifying the same semantic expression: *run*, *push*, *drag*



c. TRANSITION: an event identifying a semantic expression evaluated with respect to its opposition: *give*, *open*; *build*:
Binary transition (achievement): $\neg\phi \in S_1$, and $\phi \in S_2$



Complex transition (accomplishment): $\neg\phi \in P$, and $\phi \in S$

The basic event types are the states and processes, which can represent independent events or be combined to derive complex events (transitions). Subevents within an event are ordered by **temporal relations** and **relative prominence** or **headedness**. Regarding temporal relations, two subevent orderings are of relevance here. One subevent may precede the other in a strictly sequential relation $< \circ$ (Allen's "meet" relation (Allen, 1984)), with the first subevent leading to the second, as with causatives (e.g., *build*), inchoatives (e.g., *arrive*), and ditransitive transfer verbs (e.g., *give*). In transaction events such as *sell*, *buy*, and *marry* ('get married to'), both subevents overlap in time ($\circ$).
Notice that, unlike primitive predicates, subevents can be quantified in the logical form of the sentence, in the same way that arguments can be.

(6) a. The destroyer is sinking the boat.
$\exists e_1 \exists x \exists y[\text{sink\_act}(e_1, x, y) \wedge \text{destroyer}(x) \wedge \text{boat}(y)]$

b. The destroyer sank the boat.
$\exists e_1 \exists e_2 \exists x \exists y[\text{sink\_act}(e_1, x, y) \wedge \text{destroyer}(x) \wedge \text{boat}(y) \wedge \text{sink\_result}(e_2, y) \wedge e_1 < e_2]$

c. The boat sank.
$\exists e_2 \exists e_1 \exists y \exists x[\text{sink\_result}(e_2, y) \wedge \text{boat}(y) \wedge \text{sink\_act}(e_1, x, y) \wedge e_1 < e_2]$

The logical form of the causative (6b) differs from the inchoative (6c) only in the explicit identification of a specific causer.
In subsequent work within GL, event structure has been integrated with dynamic semantic models in order to represent the attribute modified in the course of the event (the location of the moving entity, the extent of a created or destroyed entity, etc.) as a sequence of states related to time points or intervals. This way, in addition to describing the event in terms of discrete phases, we identify what attribute is changing and how it is changing over the event. The resulting event structure representation is called a **Dynamic Event Model** (Pustejovsky and Moszkowicz, 2011; Pustejovsky, 2013). Starting with the view that subevents of a complex event can be modeled as a sequence of frames, a dynamic event model explicitly labels the transitions that move an event from frame to frame.[2] We believe that, in order to adequately model change, the VerbNet representation must track the change in the assignment of values to attributes in the unfolding of the event. This includes making explicit any *predicative opposition* denoted by the verb. For example, simple transitions (achievements) encode either an intrinsic predicate opposition (*die* encodes going from $\neg dead(e_1, x)$ to $dead(e_2, x)$), or a specified relational opposition (*arrive* encodes going from $\neg loc\_at(e_1, x, y)$ to $loc\_at(e_2, x, y)$). Creation predicates and accomplishments generally also encode predicate oppositions.
A dynamic approach to modeling updates makes a distinction between formulae, ϕ, and programs, π. A formula is interpreted as a classical propositional expression, with assignment of a truth value in a specific state in the model (Harel et al., 2001). For our purposes, a state is a set of propositions with assignments to variables at a specific frame. We can think of atomic programs as input/output

[2]The resulting structure is equivalent to a Labeled Transition System (van Benthem, 1991), and is consistent with the approach developed in (Fernando, 2009; Fernando, 2013).

relations, i.e., relations from states to states, and hence interpreted over an input/output state-state pairing (cf. (Naumann, 2001)).
The model encodes three kinds of representations: (i) predicative **content** of a frame; (ii) **programs** that move from frame to frame; and **tests** that must be satisfied for a program to apply. These include: pre-tests, while-tests, and result-tests.

5. VerbNet's New Semantic Representations

A Generative Lexicon-inspired subevent structure fulfills the need for greater expressiveness and clarity in VerbNet representations that we identified in section 3. In this section we will describe the global changes we are making and then present their application in a few of the more complex types of events in VerbNet.
The greatest change is switching from a tripartite division of the temporal span of any event to a system of numbered subevents, which can be increased or decreased to accommodate the complexity of the event. This change eliminates the second-order logic of Start(E), During(E) and End(E), which was necessary to ease the integration of the representations with a robot planning system. It also allows for more nuanced temporal relationships between the subevents, as described in section 3.
The default assumption in this new schema is that e_1 precedes e_2, which precedes e_3, and so on. When appropriate, however, more specific predicates can be used to specify other relationships, such as **meets**(e_2, e_3) to show that the end of e_2 meets the beginning of e_3, or **while**(e_2, e_3) to show that e_2 and e_3 are co-temporal. The latter can be seen in section 5.1 with the example of accompanied motion.
Another important change is the way in which causation is represented. Previously, the representation implied that an event as a whole was being caused by an Agent, using **cause**(Agent, E).

(7) *The lion tamer jumped the lions through the hoop.*
NP V NP PP
Agent V Theme Trajectory

motion(during(E), Theme)
path_rel(start(E), Theme, ?Initial_location, ch_of_loc, prep)
path_rel(during(E), Theme, Trajectory, ch_of_loc, prep)
path_rel(end(E), Theme, ?Destination, ch_of_loc, prep)
cause(Agent, E)

In the new version, we focus on one subevent as being the cause of another. Thus, something an agent does (e.g., **do**(e_2, Agent)) causes a state change or another event (e.g., **motion**(e_3, Theme)), which would be indicated with **cause**(e_2, e_3).

(8) *The lion tamer jumped the lion through the hoop.*
has_location(e_1, Theme, ?Initial_Location)
do(e_2, Agent)
motion(e_3, Theme, Trajectory)
cause(e_2, e_3)
has_location(e_4, Theme, ?Destination)

(See sections 4.1-4.3 for further examples.)
A more minor adjustment concerns the path_rel predicate, which was introduced earlier in the revision process to highlight the commonalities among different types of change events and to provide greater consistency in the existing VerbNet representations (Hwang, 2014). At the request of some users, we are substituting more specific predicates for the general **path_rel** predicate, such as **has_location**, **has_state** and **change_value**. This shifts some information that was included within the variables and constants of the **path_rel** predicate out to the new predicates themselves. We are however maintaining a common subevent pattern for change of location, change of possession and change of state events that closely mirrors that introduced by Hwang (2014).
Events that include some sort of change from one location to another or one state to another compose the majority of classes in VerbNet and include some of the more complex event types. Therefore, we will use examples from the change of state and change of location classes to illustrate the new VerbNet representations.

5.1. Change of Location

The Run-51.3.2 class is a typical change of location class, with such member verbs as *run, march,* and *gallop.* The most basic change of location semantic representation (9) begins with a state predicate **has_location**, with a subevent argument e_1, a Theme argument for the object in motion, and an Initial_location argument. The motion predicate is underspecified as to the manner of motion in order to be applicable to all 97 verbs in the class. A final **has_location** predicate indicates the Destination of the Theme at the end of the event. Not all of the thematic roles included in the representation are necessarily instantiated in the sentence. Any uninstantiated roles in a frame are preceded by a question mark, such as Initial_location and Trajectory in (9).

(9) *The rabbit hopped across the lawn.*
has_location(e_1, Theme, ?Initial_Location)
motion(e_2, Theme, Trajectory)
has_location(e_3, Theme, ?Destination)

This representation collapses the information in two semantic predicates in the old VerbNet representation (10): the **path_rel** predicate indicating the Trajectory of the Theme with the motion predicate.

(10) *The rabbit hopped across the lawn.*
motion(during(E), Theme)
path_rel(start(E), Theme, ?Initial_location, ch_of_loc, prep)
path_rel(during(E), Theme, Trajectory, ch_of_loc, prep)
path_rel(end(E), Theme, ?Destination, ch_of_loc, prep)

This pattern of an initial state followed by a transition to another state forms the basis for more complex events, such a caused change of location. The representation in (9) is augmented in (11) with both a DO and a CAUSE predicate.

(11) *The farmer herded the sheep into the meadow.*
has_location(e_1, Theme, ?Initial_Location)
do(e_2, Agent)
motion(e_3, Theme, ?Trajectory)
cause(e_2, e_3)
has_location(e_4, Theme, Destination)

For many classes, the causal action has more specific semantic components in common across the member verbs, and, therefore, the predicates can be more specific than the underspecified DO predicate in the Run-51.3.2 class. For example, the Push-12 class represents the Agent's action with **contact** and **exert_force** predicates (12).

(12) *John pushed the plate to the edge of the table.*
has_location(e_1, Theme, ?Initial_Location)
contact(e_2, Agent, Theme)
exert_force(e_2, Agent, Theme)
motion(e_3, Theme, ?Trajectory)
has_location(e_4, Theme, Destination)

This new version highlights several of the advantages we have been discussing in comparison to the old (13).

(13) *John pushed the plate to the edge of the table.*
cause(Agent, E)
contact(during(E), Agent, Theme)
exert_force(during(E), Agent, Theme)
path_rel(start(E), Theme, ?Initial_location, ch_of_loc, prep)
path_rel(during(E), Theme, Trajectory, ch_of_loc, prep)
path_rel(end(E), Theme, ?Destination, ch_of_loc, prep)
motion(during(E), Theme)

On a superficial level, the new representation is more transparent to human readers, with the starting and ending states indicated with has_location predicates, and the numbered subevents clearly stepping through the temporal sequence. More fundamentally, the numbered subevents allow us to divide what was previously grouped as During(E) into separate subevents, one involving the interaction of the Agent and Theme and one involving the motion of the Theme.
Finally, an example from the Accompany-51.7 class illustrates how the new schema represents accompanied motion (14).

(14) *Elena guided Frank through the building.*
has_location(e_1, Theme, ?Initial_Location)
has_location(e_2, Agent, ?Initial_Location)
motion(e_3, Agent, Trajectory)
motion(e_4, Theme, Trajectory)
has_location(e_5, Agent, ?Destination)
has_location(e_6, Theme, ?Destination)
while(e_3, e_4)

The predicate **while** allows us to indicate that both the Agent and Theme are in motion simultaneously.

5.2. Change of State

The representations for changes of state have two basic patterns, depending on whether the change is between absolute states or along a value continuum. The first is illustrated in (15), the representation for the Die-42.4 class.

(15) *John died.*
alive(e_1, Patient)
¬**alive**(e_2, Patient)

For less semantically coherent classes, such as the Other_cos-45.4 class, the type of state must be underspecified, as in (16). In that case, the opposition between the initial and the result states must be explicitly shown.

(16) *The balloon burst.*
has_state(e_1, Patient, Initial_State)
opposition(Initial_State, V_Result)
has_state(e_2, Patient, V_Result)

Like the underspecificity of the **do** predicate, **has_state** allows us to reference initial states and final states general enough to apply to all the verbs in a class. The **do** predicate is used in situations in which the Agent's action causes another subevent but we really can't determine what that action is without further context. In many of change of state classes, however, we can further identify the final state by extracting information from the verb itself. In (16), the verb 'burst' tells us the final state of the Patient. The same holds for the other verbs from the class, such as *dry, blacken* or *triple.* We have introduced V_Result both as an indicator that the semantic representation can be further refined in context using the lexical features of the specific verb and as a placeholder for that information.
V_Result also allows us to distinguish between the change of state introduced by the verb and a further change of state introduced by a resultative construction.

(17) *The clothes dried wrinkled.*
Theme V Result
has_state(e_1, Patient, Initial_State)
has_state(e_2, Patient, V_Result)
has_state(e_2, Patient, Result)
opposition(Initial_State, V_Result)
opposition(Initial_State, Result)

A second type of change of state involves a change along a scale, such as the events in the Calibratible_cos-45.6.1 class.

(18) *The price of oil rose by 500% from $5 to $25.*
has_val(e_1, Patient, Initial_State)
change_value(e_2, DIRECTION, Extent, Attribute, Patient)
has_val(e_3, Patient, Result)

The members of this class have verb-specific features, either increase (e.g., *rise*), decrease (e.g., *fall*) or fluctuate (e.g., *vary*). DIRECTION, one of the arguments of **change_value**, is a variable whose value can be found in context from the particular verb's verb-specific feature.

6. Conclusion

This paper has focused on specific changes to the semantic representations associated with classes of verbs in VerbNet. We have restricted the representation language to first-order representations to simplify use by and integration with planners. A larger change has been modifications to incorporate GL's event structure, with temporal ordering of subevents associated with explicit predications over the verb's arguments. This allows for greater flexibility in representing complex events, for a more consistent treatment of the oppositions inherent in change-of-state classes, and for a more nuanced portrayal of the Agent's role.

7. Bibliographical References

Abend, O., Reichart, R., and Rappoport, A. (2008). A supervised algorithm for verb disambiguation into verbnet classes. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 9–16. Association for Computational Linguistics.

Allen, J. (1984). Towards a general theory of action and time. *Arificial Intelligence*, 23:123–154.

Bonial, C., Corvey, W., Palmer, M., Petukhova, V. V., and Bunt, H. (2011). A hierarchical unification of lirics and verbnet semantic roles. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 483–489. IEEE.

Brown, S. W., Dligach, D., and Palmer, M. (2014). Verbnet class assignment as a wsd task. In *Computing Meaning*, pages 203–216. Springer.

Fernando, T. (2009). Situations in ltl as strings. *Information and Computation*, 207(10):980–999.

Fernando, T. (2013). Segmenting temporal intervals for tense and aspect. In *The 13th Meeting on the Mathematics of Language*, page 30.

Giuglea, A.-M. and Moschitti, A. (2006). Semantic role labeling via framenet, verbnet and propbank. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 929–936. Association for Computational Linguistics.

Harel, D., Kozen, D., and Tiuryn, J. (2001). Dynamic logic. In *Handbook of philosophical logic*, pages 99–217. Springer.

Hwang, J. D. (2014). *Identification and representation of caused motion constructions*. Ph.D. thesis, University of Colorado at Boulder.

Kawahara, D. and Palmer, M. (2014). Single classifier approach for verb sense disambiguation based on generalized features. In *LREC*, pages 4210–4213.

Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2008). A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1):21–40.

Loper, E., Yi, S.-T., and Palmer, M. (2007). Combining lexical resources: mapping between propbank and verbnet. In *Proceedings of the 7th International Workshop on Computational Linguistics, Tilburg, the Netherlands*.

Mani, I. and Pustejovsky, J. (2012). *Interpreting motion: Grounded representations for spatial language*. Number 5. Oxford University Press.

Narayan-Chen, A., Graber, C., Das, M., Islam, M. R., Dan, S., Natarajan, S., Doppa, J. R., Hockenmaier, J., Palmer, M., and Roth, D. (2017). Towards problem solving agents that communicate and learn. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 95–103.

Naumann, R. (2001). Aspects of changes: a dynamic event semantics. *Journal of semantics*, 18(1):27–81.

Pustejovsky, J. and Moszkowicz, J. (2011). The qualitative spatial dynamics of motion. *The Journal of Spatial Cognition and Computation*.

Pustejovsky, J., Palmer, M., Zaenen, A., and Brown, S. (2016). Verb meaning in context: Integrating verbnet and gl predicative structures. In *Proceedings of the LREC 2016 Workshop: ISA-12, Potoroz, Slovenia*, volume 2016.

Pustejovsky, J. (1995). *The Generative Lexicon*. Bradford Book. Mit Press.

Pustejovsky, J. (2005). Generative lexicon and type theory. ESSLLI Summer School, 2005, Edinburgh, Scotland, August.

Pustejovsky, J. (2013). Dynamic event structure and habitat theory. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon (GL2013)*, pages 1–10. ACL.

Schuler, K. K. (2005). Verbnet: A broad-coverage, comprehensive verb lexicon.

Shi, L. and Mihalcea, R. (2005). Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing. In *International conference on intelligent text processing and computational linguistics*, pages 100–111. Springer.

van Benthem, J. F. A. K. (1991). Logic and the flow of information.

Zaenen, A., Bobrow, D. G., and Condoravdi, C. (2008). The encoding of lexical implications in verbnet predicates of change of locations. In *LREC*.

8. Language Resource References

Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2006). Extensive classifications of english verbs. In *Proceedings of the 12th EURALEX International Congress*, Turin, Italy.