

# Computationally Constructed Concepts: A Machine Learning Approach to Metaphor Interpretation Using Usage-Based Construction Grammatical Cues

**Zachary Rosen**

Department of Linguistics

The University of Colorado, Boulder

295 UCB, Boulder, CO 80309

Zachary.P.Rosen@colorado.EDU

## Abstract

The current study seeks to implement a deep learning classification algorithm using argument-structure level representation of metaphoric constructions, for the identification of source domain mappings in metaphoric utterances. It thus builds on previous work in computational metaphor interpretation (Mohler et al. 2014; Shutova 2010; Bollegala & Shutova 2013; Hong 2016; Su et al. 2017) while implementing a theoretical framework based off of work in the interface of metaphor and construction grammar (Sullivan 2006, 2007, 2013). The results indicate that it is possible to achieve an accuracy of approximately 80.4% using the proposed method, combining construction grammatical features with a simple deep learning NN. I attribute this increase in accuracy to the use of constructional cues, extracted from the raw text of metaphoric instances.

## 1 Introduction

Lakoff's theory of conceptual metaphor has been highly influential in cognitive linguistic research since its initial publication (Lakoff & Johnson 1980). Conceptual metaphors represent fine-grained mappings of abstract concepts like "love" to more concrete, tangible phenomena, like "journeys" which have material and culturally salient attributes like a PATH, various LANDMARKS, and a THEME which undergoes movement from a SOURCE to a GOAL (Lakoff & Johnson 1980). These tangible phenomena then serve as the basis for models from which speakers can reason about abstract ideas in a culturally transmissible manner. For example, consider the following metaphoric mappings for the metaphor LOVE IS MAGIC, as shown in figure 1.

To date, while automatic metaphor detection has been explored in some length, computational

metaphor interpretation is still relatively new, and a growing number of researchers are beginning to explore the topic in greater depth. Recently, work by the team behind Berkeley's MetaNet has shown that a constructional and frame-semantic ontology can be used to accurately identify metaphoric utterances and generate possible source domain mappings, though at the cost of requiring a large database of metaphoric exemplars (Dodge et al. 2015; Hong 2016). Researchers from the Department of Cognitive Science at Xiamen University (Su et al. 2017) report that, using word embeddings, they have created a system that can reliably identify nominal-specific conceptual metaphors as well as interpret them, albeit within a very limited scope—the nominal modifier metaphors that they work with only include metaphors in which the source and target domain share what they refer to as a "direct ancestor", such as in the case of "the surgeon is a butcher", limiting researchers to analyzing noun phrases with modifiers that exist in a single source and target domain. Other approaches have included developing literal paraphrases of metaphoric utterances (Shutova 2010; Bollegala & Shutova 2013), and, as an ancestor to the current study, clustering thematic co-occurents—the AGENT, PATIENT, and ATTRIBUTE of the metaphoric sentence—which allowed researchers to predict a possible source domain label—think: "The bill blocked the way forward", where for the word "bill" the system predicted that it mapped to a "PHYSICAL OBJECT" role in the source domain (Mohler et al. 2014).

## 2 Construction Grammatical Approaches to Metaphor

The constructional makeup of metaphoric language has been explored at some length by a

LOVER is a MAGICIAN	She cast her spell over me
ATTRACTION is a SPELL	I was spellbound
A RELATIONSHIP is BEWITCHMENT	He has me in a trance

Figure 1: Metaphoric Mapping & Example

handful of researchers to date. Karen Sullivan, for example, has done considerable work on both how syntactic structures (i.e. constructions) restrict the interpretation of metaphoric utterances in predictable ways by both instantiating a semantic frame and mapping the target domain referent to a semantic role *within* the instantiated frame (Sullivan 2006, 2009, 2013). Notable examples of computational implementations of Sullivan’s theories include Stickles et al. (2016) and Dodge et al. (2015), who have compiled a database of metaphoric frames—MetaNet—organized into an ontology of source domains for researchers to use in analyzing metaphoric utterances, similar to FrameNet.

One of the advantages of construction grammar with respect to figurative language interpretation lies in the regularity with which constructions establish form-meaning pairings. The various meanings of constructions rely heavily on particular “cues”—cues including the verb, as well as the syntactic template and argument-structure—which point speakers in the direction of a specific interpretation (Goldberg 2006). For the purpose of the current study, I will be focusing on the argument-structure of metaphoric utterances which, though it supplies a rather course-grained view of the meaning of an utterance, provides an excellent and stable constructional cue with respect to its interpretation (Goldberg 2006). As an example of how this might work, consider the difference between “the Holidays are coming up on us” and “we’re coming up on the Holidays.” In the first sentence, “the Holidays” is established as being mapped to a MOVING OBJECT in the source domain by virtue of its position in the argument-structure of the sentence. Meanwhile, in the second utterance “the Holidays” is mapped to a LOCATION or GOAL in the source domain due to its change in position in the argument-structure of the construction. Implicitly, this means that important information about the interpretation of a construction can be gleaned through extracting the arguments that fill its argument-structure and analyzing these arguments’ relationships to one another,

independent of cues beyond the sentence itself.

### 3 Data Collection

All the examples in this experiment were taken from the EN-Small LCC Metaphor Dataset, compiled and annotated by Mohler et al. (2016). The corpus contains 16,265 instances of conceptual metaphors from government discourse, including immediate context sentences preceding and following them. Each sentence is given a metaphoricity score, ranging from “-1” to “3”, where “3” indicates high confidence that the sentence is metaphoric, “0” indicates that the sentence was not metaphoric, and “-1” indicates an invalid syntactic relationship between the target and source domain referents in the sentence (Mohler et al. 2016). Additionally, the corpus is annotated for polarity (negative, neutral, and positive), intensity, and situational protagonists (i.e.: the “government”, “individuals”, etc.). Though not annotated for every sentence, the most important annotations for this study were the annotations for source-target domain mappings. There was a total of 7,941 sentences annotated for these mappings, with 108 source domain tags, annotated by five annotators (Mohler et al. 2016). Each annotator indicated not only what they thought the source domain was, but also gave the example an additional metaphoricity score based on their opinion.

For the purposes of this study, I only used the metaphoric instances that were annotated for source-target domain mappings. For the source domain labels, I selected the labels made by the annotator who had marked the example for having the highest metaphoricity. I initially attempted to select the metaphoric source domain annotations that had the highest agreement amongst the annotators who had annotated the sentence, but this proved trickier than I had anticipated. After calculating the average Cohen Kappa score (54.4%), I decided that selecting labels based on their associated metaphoricity would be better. This effectively removed two annotators from the pool, who consistently ranked each metaphoric sentence as having a metaphoricity score of 1 or less.

I further restricted the training and test data by excluding multi-word expressions from the dataset for simplicity, though in the future I would very much like to re-test the methods outlined in the rest of this paper including the omitted MWEs. Finally, I removed any source domain annotations that included only a single example and split the data in training and testing data sets, using 85% as training data, and 15% as testing data. Because of my exclusion of MWEs and metaphoric source domain tags that were used only once, this left me with a total of 1985 sentences used in this experiment—1633 of those were used in the training data, and 352 reserved for test data—with 77 source domain labels. The source labels were converted to integers and used as classes in the following Deep Neural Net (DNN) classifier.

## 4 The Neural Network Approach to Source Domain Interpretation

### 4.1 Feature Generation

The task in this study is to predict the source domain of a metaphoric utterance using only features extracted from the sentence text. For example, from a sentence like "So, you advocate for the ability to deny people the vote by pushing them into poverty?", and given the target domain referent (in this sentence, "poverty"), can we accurately predict the source domain label "ABYSS" (as annotated in the LCC dataset) using only the text from the sentence? To do so, we wanted to extract from the sentence a representation of its argument structure, and use that to classify the source domain label. The argument structure of a construction is represented by the verb and the arguments it accepts to fulfill the roles defined by both the verb and its semantic frame (Goldberg 2006; Michaelis 2012; Sag 2012; Pustejovsky 2011). Though there are subtle differences between construction grammar and dependency grammar, it is possible to reconstruct the argument-structure of a construction from grammatical dependencies (Osborne & Gross 2012; for a computational implementation of a theoretically similar system to ours, see Hong 2016). For the purposes of this study, I first generated a representation of all the dependency relationships in each sentence from the LCC dataset using the Stanford NLP dependency parser (Chen & Manning 2014). Second, I searched the output list dependencies from the dependency parser for the target domain referent as identified in the

corpus example, and found the verb that it was directly dependent on in the sentence. This ensured that the target domain referent was in its immediate context. Once the verb was found, I then built a representation of the argument structure of the sentence by extracting the following dependencies—(1) the verb for which the target domain referent was a dependency, (2) the subject of the verb in 1, (3) the object of the verb in 1, and if the target domain referent was not included in the subject or direct object, (4) the target domain referent as a nominal modifier and (5) any prepositional arguments that it had as a dependency. Additionally, I extracted (6) the universal dependency tags for each of the arguments in the verb's argument-structure, and converted that into a list of tags that I simply labeled "syntax", or "SYN", based off the assumption that knowing what the dependencies were might help in identifying the exact relationships between the lexemes that had been collected. Finally, these elements along with (7) the target domain referent itself were compiled into a list to be used in the training or test data, and labeled with the pre-identified source domain label assigned to the sentence in the LCC dataset. The output of this process is visually represented in figure 2. The branch of the dependency tree in blue indicates the direct context of the target domain referent—in this case, "poverty".

While these strings provided a representation of the arguments as a set, they did not provide enough information *a priori* to predict the source domain on their own. Sullivan (2013) explains that the backbone of metaphoric utterances is the relationship of the target domain referent to the frame evoked by the construction. Additionally, Goldberg (2006) describes the semantic meaning of constructions as arising from both the nouns contained in their argument-structure, and the meaning implied by the construction's syntactic template. The following features combined Sullivan's relationships of the target domain referent to the construction, with the two observations made by Goldberg about constructional meaning. For the interaction of *the target domain referent with the nouns contained in the argument structure* I used the following interactions as features: (8) the target domain referent and the subject of the local dependency tree (again, in blue in figure 2), (9) the target domain referent and the direct object, and (10) the target domain referent and the nom-

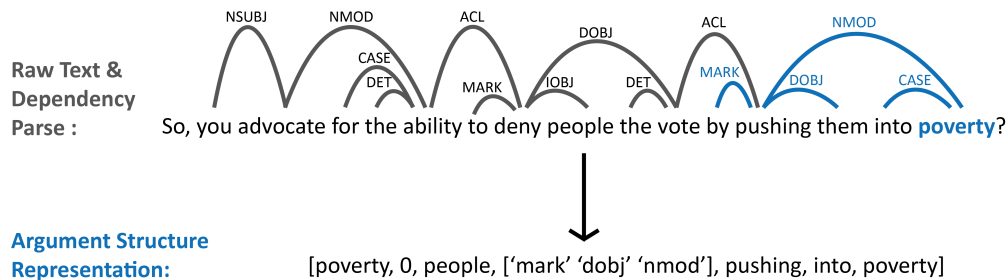


Figure 2: Raw text sentence input to dependency parsed output from Stanford Core NLP’s dependency parser. The syntactic roles in order of their index in the dependency parsed sentence are as follows: (0) the target domain referencing noun, (1) the subject, (2) the direct object, (3) the syntactic sisters of the target domain referencing noun, (3) the verb, (4) the preposition/case of a nominal modifier, (5) the head of a nominal modifier. The word in bold was annotated as the target domain referent by annotators.

inal modifier from 4 in the previous paragraph. I then augmented these with the following interactions to represent the interaction of *the target domain referent with the syntactic template*: (11) the target domain referent, the verb, and the subject of the verb, (12) the target domain referent, the verb, and the object of the verb, and (13) the target domain referent, the preposition preceding the nominal modifier, and the nominal modifier. I predicted that these six interactions would approximate the relationship between the target domain referent and its construction-based context, as inspired by previous work in semantic role labeling (Wang et al. 2009; Matsubayashi et al. 2014; and especially Gildea & Jurafsky 2002, where researchers automatically labeled the semantic role of a specific target noun in a given frame). A list of these complex interactions can be seen in figure 3.

These 13 features were then converted into embeddings to be used as inputs in the DNN via the following process. The strings extracted from the dependency parsed, raw text sentence were first lemmatized, then converted from strings into numeric representations in Tensorflow using the `tf.contrib.layers.sparse column with hash bucket` function. The interactions indicated in 8-13 in the prior paragraph were defined using the `tf.contrib.layers.crossed column` function, returning a numeric representation of the interaction. Finally, these numeric representations for all of the features described above were then converted into an embedding layer in order to represent the context of the features as they appeared per each sentence that they extracted from. This was done us-

ing the `tf.contrib.layers.embedding column` function, and the number of dimensions for each embedding layer was set uniformly at 13 dimensions.

## 4.2 Feed Forward DNN Network Architecture

These embedding layers were then used as the inputs into the DNN. In order to quickly prototype the model, I used the `tf.contrib.learn` library in Tensorflow. The activation function in the network was set to a `relu` function (`tf.nn.relu`). The network included a single, fully connected hidden layer, with 77 hidden units which were randomly initialized during training. I implemented a dropout rate of .4 during training to prevent overfitting. Information from the hidden layer was passed to a Softmax layer, and then passed to an output layer for the 77 labels in the train and test data. The reason behind using a single hidden layer was in part because the model training was initially done on a single MacBook Air, and so the model needed to be sufficiently small to train efficiently on that computer. The network was trained for 500 epochs, or until the model reached a training loss less than .006 after the 498th epoch. The early cut-off was decided upon after having run the model 20 times, and having discovered that accuracy was improved by approximately 1.2% if training was cut off immediately after reaching a loss less than .006. The full network architecture can be seen in figure 4.

## 4.3 Accuracy and Evaluation

The DNN architecture as described accurately predicted the source domain label from the LCC

<b>Argument Structure Representation:</b>	[poverty, 0, people, ['mark' 'dobj' 'nmod'], pushing, into, poverty]
<b>Target Domain NP x Subject:</b>	[poverty, 0]
<b>Target Domain NP x Direct Object:</b>	[poverty, people]
<b>Target Domain NP x Head of Nominal Modifier:</b>	[poverty, poverty]
<b>Target Domain NP x Verb x Subject:</b>	[poverty, pushing, 0]
<b>Target Domain NP x Verb x Direct Object:</b>	[poverty, pushing, people]
<b>Target Domain NP x Preposition x Head of Nominal Modifier:</b>	[poverty, into, poverty]

Figure 3: Diagram of the interactions as derived from the previous dependency parsed inputs.

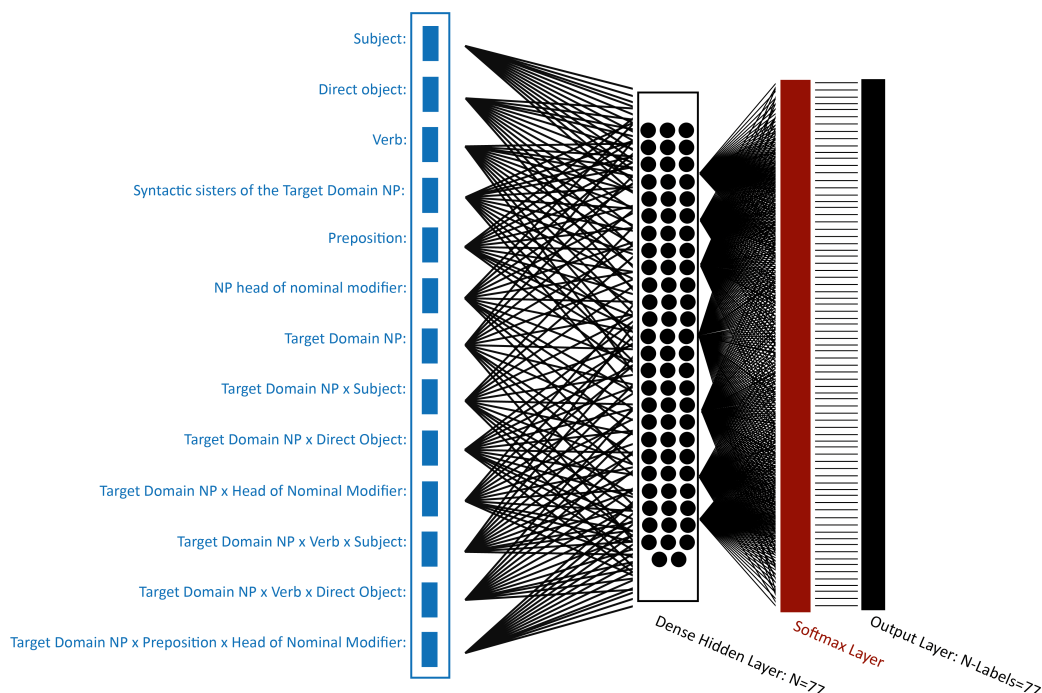


Figure 4: Diagram of the full DNN architecture from input features to output layer.

dataset 80.4% of the time, with a testing loss value of 1.51. I compared the output of the feedforward network to a similar DNN build without the interactions from figure 3 (essentially, only using the extracted argument structure as seen in figure 2). I then also compared the DNN architecture with the interactions in figure 3, to an LSTM neural network without those same constructional features. The results for the highest and lowest accuracy in a set of five test runs for each of these networks are compared in figure 5.

## 5 Discussion

The results reported indicates that the addition of construction grammatical relations to the feature set used by deep learning algorithms significantly increases the accuracy of metaphoric source domain prediction tasks.

Whilst the inclusion of the lexical units from the dependency parsed sentence are important to build sufficient context for the DNN classifier, the interactions as seen in Figure 3 provide the real predictive power of this system by approximating the relationship between the target domain referent and the interactions of items in the argument-structure of the construction. While we can take for granted from work in both VerbNet and FrameNet (VerbNet: Kipper, Korhonen, Ryant & Palmer 2008; FrameNet: Fillmore et al. 2001; Fillmore, Johnson, & Petruck 2002) proving that the verb is a strong cue for the semantic frame, a stronger predictor for the metaphoric source domain is the interaction of the verb with the arguments in its argument-structure.

In theory, the pipeline from dependencies, to usage-based constructional features, to embeddings for input into the DNN described, would

Network Architecture	Lowest acc	Highest acc
Feed Forward DNN without constructional cues as inputs	77.6%	78.9%
LSTM without constructional cues as inputs	72.1%	73.0%
Feed Forward DNN with constructional cues as inputs	79.2%	80.4%

Figure 5: Highest and lowest accuracy for the three network builds.

appear to assume that the utterance being analyzed has already been identified as metaphoric. In practice, by focusing on the relationship of the target domain referent to a small set of interactions (representing a construction’s argument-structure), one could feasibly use a known set of target domain referents in order to identify the source domains that they are mapped to, skipping entirely the need to identify an example as metaphoric. Think of it like this: if a researcher is interested in the kinds of metaphors used to talk about “poverty” in a text, a simple query coupled with the DNN described can find and accurately predict possible source domain labels for all utterances in which “poverty” is used. Coupling the DNN here with a system designed to identify metaphors or even target domain referents in a text, however, would be ideal, and would greatly add to the described DNN’s power and utility as a predictive tool.

An additional confound limiting the final accuracy in this experiment was the wide range of conceptual metaphor source domain annotations given by annotators per each utterance in the LCC dataset. Despite it being an excellent resource for researchers interested in metaphor source domain interpretation due to its CMSource annotations, the average inter-annotator agreement for source domain mappings in the corpus was on average approximately 54.4% for the dataset, as calculated by averaging the Cohen-Kappa scores for annotators. While annotators agreed about the relatedness of the source and target domain referents during the annotation process (agreement for “Source Relatedness” and “Target Relatedness” in the LCC dataset were calculated as of 2014 as 95.1% and 94.3% respectively (Mohler et al. 2014)), several of the source domain mappings provided were different from one another in incredibly subtle, but crucial, ways. Take “LMInstance” 22920 from the dataset for example—“This prison is the prison of poverty.” Where as one of the annotators labeled the sentence as evoking “CRIME” as the source domain mapping, another indicated that it evoked

the thematically related concept of “CONFINEMENT” as the source domain. Neither label in this instance appears, at least on first glance, to be intrinsically better than the other.

Adding to this, I actively avoided using examples in which MWEs were identified as the target domain referent—a decision which limited the number of examples used, and thus likely limited the number of times that a specific argument-structure construction in the dataset showed up alongside of an accompanying source-domain label.

In all, the current experiment serves as an example not only of the usefulness of construction grammar to NLP tasks, but of the utility of a cognitive theory of language understanding to computational linguistic inquiry.

## 6 Acknowledgements

I would like to thank the anonymous reviewers for their excellent feedback, and Michael Mohler of the Language Computer Corporation for the corpus used in this paper. I would also like to thank the wonderful faculty and students at the University of Colorado, Boulder, for their support.

## References

- Danushka Bollegala and Ekaterina Shutova. 2013. Metaphor interpretation using paraphrases extracted from the web. *PloS ONE* 8(9):e74304.
- Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP 2014*. Association for Computational Linguistics, pages 740–750.
- Ellen Dodge, Jisup Hong, and Elise Stickles. 2015. Metanet: Deep semantic automatic metaphor analysis. In *Proceedings of the Third Workshop on Metaphor in NLP*. Association for Computational Linguistics, pages 40–49.
- John Feldman and S. Narayanan. 2004. Embodied meaning in a neural theory of language. *Brain and language* 89(2):385–392.
- Charles Fillmore, Christopher Johnson, and Miriam Petruck. 2002. Background to framenet. *International Journal of Lexicography* 16(3):235–250.

- Charles Fillmore et al. 2001. Building a large lexical databank which provides deep semantics. In Benjamin Tsou and Olivia Kwong, editors, *Proceedings of the 15th Pacific Asia Conference on Language, Information and Computation*. Pacific Asia Conference on Language, Information, and Computation, pages 3–26.
- Dedre Gentner et al. 2002. As time goes by: Evidence for two systems in processing spacetime metaphors. *Language and Cognitive Processes* 17(5):537–565.
- Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Linguistics* 28(3):245–288.
- Adele Goldberg. 2006. *Constructions at Work: The Nature of Generalization in Language*. Oxford University Press, Inc.
- Stefan Gries. 2006. Corpus-based methods and cognitive semantics: The many senses of to run. In Stefan Th. Gries and Anatol Stefanowitsch, editors, *Corpora in Cognitive Linguistics: Corpus-Based Approaches to Syntax and Lexis*, Walter de Gruyter, pages 57–99.
- Jisup Hong. 2016. Automatic metaphor detection using constructions and frames. *Constructions and Frames* 8(2):295–322.
- Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of english verbs. *Language Resources and Evaluation Journal* 42(1):21–40.
- George Lakoff. 1990. *Women, Fire, and Dangerous Things*. The University of Chicago Press.
- George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. The University of Chicago Press.
- R. W. Landacker. 2002. *Concept, image and symbol: cognitive basis of grammar*. Mouton de Gruyter.
- R. W. Langacker. 1997. Constituency, dependency, and conceptual grouping. *Cognitive Linguistics* 8(1):1–32.
- Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics*. Association for Computational Linguistics, volume 2, pages 768–774.
- Z.J. Mason. 2004. Cormet: A computational, corpus-based conventional metaphor extraction system. *Computational Linguistics* 30(1):23–44.
- Yuichiroh Matsubayashi et al. 2014. Generalization of semantic roles in automatic semantic role labeling. *Journal of Natural Language Processing* 21(4):841–875.
- Matthew McGlone. 1998. Back (or forward?) to the future: The role of perspective in temporal language comprehension. *Journal of Experimental Psychology Learning Memory and Cognition* 24(5):1211–1223.
- Laura Michaelis. 2012. Making the case for construction grammar. In Hans Boas and Ivan Sag, editors, *Sign-Based Construction Grammar*, Center for the Study of Language and Information, pages 31–67.
- Michael Mohler et al. 2013. Semantic signatures for example-based linguistic metaphor detection. In Ekaterina Shutova, Beata Beigman Klebanov, Joel Tetreault, and Zornitsa Kozareva, editors, *First Workshop on Metaphor in NLP*. Association for Computational Linguistics, pages 27–35.
- Michael Mohler et al. 2014. A novel distributional approach to multilingual conceptual metaphor recognition. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Association for Computational Linguistics, pages 1752–1763.
- Michael Mohler et al. 2016. Introducing the LCC metaphor datasets. In *Proceedings of the Language Resources and Evaluation Conference 2016*. European Language Resources Association (ELRA), pages 4221–4227.
- James Pustejovsky. 2011. Coercion in a general theory of argument selection. *Linguistics* 49(6):1401–1431.
- Ivan Sag. 2012. Sign-based construction grammar: An informal synopsis. In Hans Boas and Ivan Sag, editors, *Sign-Based Construction Grammar*, Center for the Study of Language and Information, pages 61–197.
- Ekaterina Shutova. 2010. Automatic metaphor interpretation as a paraphrasing task. In Ben Hachey and Miles Osborne, editors, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1029–1037.
- Elise Stickles et al. 2016. Formalizing contemporary conceptual metaphor theory: A structured repository for metaphor analysis. *Constructions and Frames* 8:166–213.
- Chang Su et al. 2017. Automatic detection and interpretation of nominal metaphor based on the theory of meaning. *Neurocomputing* 219:300–311.
- Karen Sullivan. 2007. Metaphoric extension and invited inferencing in semantic change. *Culture, Language and Representation, Special Issue: Metaphor and Discourse* pages 257–274.
- Karen Sullivan. 2009. Grammatical constructions in metaphoric language. In Barbara Lewandowska-Tomaszczyk and Katerina Dziwirek, editors, *Studies in Corpus Linguistics*, John Benjamins, pages 1–24.

Karen Sullivan. 2013. *Frames and Constructions in Metaphoric Language*. John Benjamins.

Hong-Lin Wang et al. 2009. Semantic role labeling based on dependency relationship. *Computer Engineering* 35(15):82–84.