

# Leveraging Syntactic Constructions for Metaphor Identification

**Kevin Stowe**

University of Colorado, Boulder  
kevin.stowe@colorado.edu

**Martha Palmer**

University of Colorado, Boulder  
martha.palmer@colorado.edu

## Abstract

Identification of metaphoric language in text is critical for generating effective semantic representations for natural language understanding. Computational approaches to metaphor identification have largely relied on heuristic based models or feature-based machine learning, using hand-crafted lexical resources coupled with basic syntactic information. However, recent work has shown the predictive power of syntactic constructions in determining metaphoric source and target domains (Sullivan, 2013). Our work intends to explore syntactic constructions and their relation to metaphoric language. We undertake a corpus-based analysis of predicate-argument constructions and their metaphoric properties, and attempt to effectively represent syntactic constructions as features for metaphor processing, both in identifying source and target domains and in distinguishing metaphoric words from non-metaphoric.

## 1 Metaphor Background

Metaphor can be understood as the conceptualization of one entity using another. Lakoff and Johnson’s seminal work shows that metaphors are present at the cognitive level and expressed linguistically (Lakoff and Johnson, 1980). A typical conceptual metaphor mapping is ARGUMENT IS WAR, in which ARGUMENT is structured through the domain of WAR:

1. He *defended* his position through his publications.
2. Her speech *attacked* his viewpoint.

The term “linguistic metaphor” is used to indicate these types of words and phrases. We will focus on linguistic metaphor, as identifying these utterances as metaphoric is critical for generating

correct semantic interpretations. For instance, in the examples above, literal semantic interpretations of ‘defend’ and ‘attack’ will yield nonsensical utterances: a physical position cannot reasonably be defended by a publication, nor can a speech physically attack any kind of entity.

Automatic metaphor processing tends to involve two main tasks: identifying which words are being used metaphorically (here called *metaphor identification*), and attempting to provide an accurate semantic interpretation for an utterance (here called *metaphor interpretation*). The first has largely been approached as a supervised machine learning problem, typically using lexical semantic features and their interaction with context to learn the kinds of situations where lexical metaphors appear. The problem of metaphor interpretation is more complex, with approaches including the implementation of full metaphoric interpretation systems (Martin, 1990), (Ovchinnikova et al., 2014), identification of source and target domains (Dodge et al., 2015), developing knowledge bases (Gordon et al., 2015), and providing literal paraphrases to metaphoric phrases (Shutova, 2010), (Shutova, 2013).

In both identification and interpretation systems, syntax tends to play a limited role. Many systems rely only on lexical semantics of target words, or use only minimal context or dependency relations to help disambiguate in context (Gargett and Barnden, 2015), (Rai et al., 2016). Others rely on topic modeling and other document and sentence level features to provide general semantics, and compare the lexical semantics to that, ignoring the more “middle”-level syntactic interactions (Heintz et al., 2013). While these approaches have been effective in many areas, there is evidence that figurative language is significantly influenced by syntactic constructions, and thus if they can be represented more effectively, metaphor processing

capabilities can be improved.

We will examine five kinds of predicate-argument constructions in corpus data to assess their metaphoric distributions and usefulness as features for classification. Our contribution is twofold. First, we examine the LCC metaphor corpus, which includes source and target annotations, to determine their use in predicate-argument constructions (Mohler et al., 2016), and employ syntactic representations as features to improve source/target classification. Second, we investigate predicate-argument constructions in the VUAMC corpus of metaphor annotation (Praggle-jaz Group, 2007), and employ syntactic features to predict metaphoric vs non-metaphoric words.

## 2 Metaphor and Constructions

Recent metaphor research has indicated that construction grammar can be employed to determine the source and target domains of linguistic metaphors (Sullivan, 2013). In many cases, certain constructions can determine what syntactic components are allowable as source and target domains. For example, verbs tend to evoke source domains. The target domain is then evoked by one or more of the verb’s arguments (from Sullivan pg 88):

1. the **cinema** *beckoned* (intransitive)
2. the **criticism** *stung* him (transitive)
3. Meredith *flung* him **an eager glance** (ditransitive)

In these instances, the verb is from the source domain and at least one of the objects is from the target. However, arguments can also be neutral and don’t necessarily evoke the target domain. Pronouns like ‘him’ in (2) and (3) don’t evoke any domain. The optionality of domain evocation makes it harder to predict which elements of the construction participate in the metaphor. Despite this limitation, this analysis shows that syntactic structures beyond the lexical level can be indicative of source and target domains. To better understand how these structures determine metaphor, we explored metaphor-annotated corpus data for predicate-argument constructions.

## 3 Computational Approaches

While metaphor processing has largely been focused on capturing lexical semantics, there have

been a variety of approaches that incorporate syntactic information. Many computational approaches focus on specific constructions, perhaps indicating the need to classify different metaphoric constructions through different means. The dataset of (Tsvetkov et al., 2014) provides adjective-noun annotation which has been extensively studied (Rei et al., 2017), (Bulat et al., 2017). A particularly promising approach is that of (Gutierrez et al., 2016), who use compositional distributional semantic models (CDSMs) to represent metaphors as transformations in vector space, specifically for adjective-noun constructions. Another relevant approach is that of (Haagsma and Bjerva, 2016) who use clustering and selectional preference information to detect metaphors in predicate argument constructions, including verbs with objects, subjects, and both. Their highest F1 is 57.8 for verbs with both arguments.

Many systems that rely heavily on lexical resources also include some dependency information. (Rai et al., 2016) and (Gargett and Barnden, 2015) use a variety of syntactic features including lemma, part of speech, and dependency relations. However, both systems are feature-rich and these syntactic elements’ contribution is unclear. (?) use lexical features along with contrasting those features between the target word and its head. (Dodge et al., 2015) employ a variety of constructions in identifying metaphoric source and target domains. They identify a broad range of constructions and use these as templates that metaphoric expressions can fill. Our work expands on this idea by formalizing the constructions into features for statistical metaphor identification.

Perhaps the most syntactically oriented metaphor identification system is that of (Hovy et al., 2013), who uses syntactic tree kernels to identify metaphor. They use combinations of syntactic features via tree kernels and semantics via WordNet supersenses and target word embeddings. Our approach expands on this by exploring different syntactic representations and incorporating semantics through word embeddings into the syntactic structures.

## 4 Corpus Analysis

Sullivan identifies a large number of constructions and the possible configurations of their arguments with regard to source and target domains. While some corpus examples are provided that show the

variety of source-target patterns in each construction’s argument structure, an in-depth analysis of how these constructions and their metaphoric properties are distributed is still needed. We examined the predicate argument constructions they analyze by using hand-annotated metaphor corpora to better understand the distributional patterns that occur. This allows us to make predictions about what kind of constructions and arguments are useful for metaphor identification and interpretation and what might be a computationally feasible way to implement them.

While they examine many kinds of constructions, most of them seem based almost entirely on the lexical semantics of the words involved, and thus can be captured simply by effectively representing the meaning of individual words. Domain and predicative adjective constructions fall into this category: the construction is identified by the type of adjective, which needs to be represented at the lexical level. The more interesting cases are argument structure constructions, which take many forms. Sullivan identifies nine different argument structure constructions that each have their own source and target properties:

1. **Intransitive**
2. **Transitive**
3. Intransitive Resultative
4. Transitive Resultative
5. **Ditransitive**
6. **Equation**
7. Predicative AP
8. Predicative PP
9. **Simile**

To identify the use of metaphor in these constructions, we will rely on two resources: the LCC metaphor corpus and the VUAMC corpus. The freely available portion of the LCC corpus contains approximately 7,500 source/target pairs, allowing for a more in-depth look at metaphoric semantics. The VUAMC contains approximately 200,000 words of text with each word tagged as metaphoric or non-metaphoric. This allows for large scale analysis of metaphoricity versus non-metaphoricity at the word level.

#### 4.1 Identifying Constructions

To examine metaphors in these corpora, we need a method for automatically identifying predicate-argument constructions. The VUAMC corpus, as a subsection of the BNC baby, comes with

gold-standard dependency parses. For the LCC dataset, we used the dependency parser from Stanford Core NLP tools (Manning et al., 2014). These parses are sufficient to identify intransitives, transitives, and ditransitive constructions. Verb instances that have an indirect object are ditransitive, those that lack an indirect object but have a direct object are transitive, and those that lack either but have a subject are intransitive. Copulas are marked in the dependency parses, so we can easily identify equative constructions. While similes can take many forms, Sullivan’s work focuses on simile constructions that consist of a copular verb and the word ‘like’. This oversimplifies to some degree, as many similes don’t need a copula (‘she fretted like a mother hen’, ‘they flew like bats’), but it allows us to create a subset of equative constructions that represent copular similes.

This analysis is necessarily limited, as we cannot automatically capture more complex constructions via dependency parses, and many of these are often metaphorically rich. While we understand this limitation, we believe that we can utilize syntactic features of these basic constructions as a starting point, with a future goal of expanding to more complex examples.

Also note that we only identify the surface realization of these constructions - any dropped arguments or missing elements that aren’t in the dependency parse aren’t considered a part of the construction. Thus we see examples of typically ditransitive verbs (like ‘give’) that occur intransitively and transitively, as they lack overt direct and indirect objects.

## 5 LCC Analysis

To explore source and target domains, we employ the free portion of the LCC corpus from Mohler et al, which contains approximately 7,500 source/target metaphor pairs in sentential context, rated from 0 to 3 on their degree of metaphoricity. For our research, we included only those instances that were rated above 1.5, yielding approximately 3,000 metaphoric sentences. These annotations also include the source and target domains of the metaphors, and the lexical trigger phrases that engender the source and target domains. This allows us to quantify Sullivan’s analysis of source and target domains in different constructions, and shows the actual distribution of source and target domain items in each construction.

In order to identify constructions in the LCC data, we extracted syntactic relations from the dependency parses, using the basic patterns previously defined to identify predicate argument constructions. This allows us to identify the five different constructions: intransitives, transitives, ditransitives, equatives (copulas), and similes (analyzed as a subset of equative constructions). For each construction found, we can identify the predicate and the predicate’s arguments, and determine for each whether they are identified as metaphoric and whether they belong to the source or target domain.

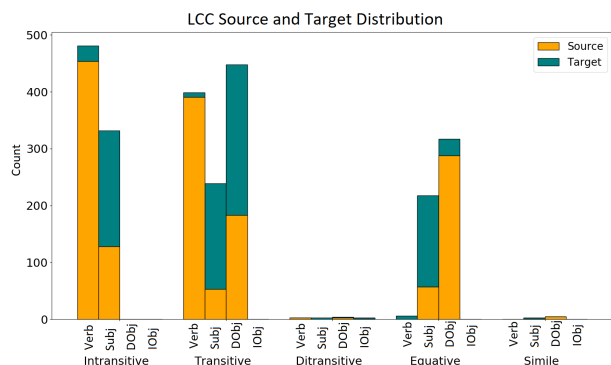


Figure 1: Counts of metaphoric items in the LCC. Each bar represents the total instances of argument in each construction, as well as the percentage of items that belong to source and target domains.

The vast majority of constructions in the LCC are intransitive, transitive, and equative. Ditransitives (.4%) and similes (.1%) are exceedingly rare. This may be because the similes found are only the verbal type: instances of a copula with the word ‘like’. Other similes are likely missed by this automatic approach.

The majority of metaphoric verbs (92%) are source domain items, supporting Sullivan’s claims. Subjects and objects tend to be from the target domain (61% each). Ditransitive verb constructions are relatively rare, with only 43 found, and only 3 of those containing a metaphoric verb.

Figure 1 shows the counts of source and target items in the LCC data, based on construction and argument of the construction. Note that in equative constructions, direct objects are almost always source domain items, showing a parallel between copular arguments and verbs. This is likely due to the predicative nature of the direct objects of copular verbs.

## 5.1 Source and Target Identification

Given that verbs and their argument structures have varying distributions of source and target domain items, we believe that these syntactic structures can be effectively employed in the classification of source and target domain words. While identifying source and target domains at the sentence level requires lexical and sentential semantics and may not require syntactic information, identifying lexical triggers can be improved by using better syntactic representations. To this end we set up a classification task for identifying source and target elements.

The LCC contains phrase-level annotations for source and target elements. We split each sentence into words, projecting the source and target annotations to the word level. From this, we developed three classification tasks: (1) identifying source words, (2) identifying target words, and (3) identifying any metaphoric word (either source or target). Our classification scheme focuses on verbs and nouns, as these are the elements that compose the syntactic structures in question.

We developed a set of different representations designed to capture construction-like structures, and employ them for source/target classification. This approach follows the intuition of (Hovy et al., 2013): “metaphorical use differs from literal use in certain syntactic patterns”. We implemented this theory by developing various representations of constructional syntax and pairing them with lexical semantic features.

For our lexical semantics component, we experimented with the word embeddings from word2vec (Mikolov et al., 2013), using the pre-trained Google News data, as well as the Glove embeddings (Pennington et al., 2014). We found in validation that the Google News vectors yielded slightly better performance, and so those were used in further experiments.

## 5.2 Syntactic Representations

Hovy et al use tree kernels to represent the semantic structure of instances, providing information from dependency parses, part of speech tags, and WordNet supersenses. Our approach follows this work by experimenting with a variety of different ways of meshing syntactic and semantic components. This involves creating a computationally feasible syntactic representation and combining it with semantics (in our case, word embeddings)

Construction	%	Verb			Subject			Direct Object			Indirect Object		
		SRC	TRG	-MET	SRC	TRG	-MET	SRC	TRG	-MET	SRC	TRG	-MET
Intransitive	66.5	454	24	6329	128	204	2385	-	-	-	-	-	-
Transitive	20.0	391	8	1648	53	186	1808	183	265	1599	-	-	-
Ditransitive	.4	3	0	40	0	3	40	3	1	39	1	2	40
Equation	13.0	0	6	1323	57	161	909	288	29	1012	-	-	-
-Simile	.1	0	0	13	1	2	10	8	5	0	-	-	-

Table 1: % Metaphor by Construction (LCC). For each predicate, the count of source (SRC), target (TRG), and non-metaphoric (-MET) instances are counted, as well as those for all of each construction’s defining arguments.

from relevant contexts.

### 5.2.1 Predicate Argument Construction

For a basic integration of syntax, we used the above corpus analysis technique to identify which predicate-argument construction the verb token belongs to. This results in a one-hot vector representing either an intransitive, transitive, ditransitive, equative, or simile construction. This provides basic, purely syntactic knowledge of how many arguments this particular instance of a verb currently has. For nouns, we extend this to include which slot in the construction the noun is filling (subject, direct object, indirect object) in addition to the type of predicate-argument construction.

### 5.2.2 Head and Dependent Features

Including representations of the head word and dependent words of the word to be classified is a straightforward way to include basic syntactic information. For verbs, this mainly involves the dependents, although many verbs also have head words. We include a concatenation of the average embedding over the word’s dependents and the embedding of the word’s head.

### 5.2.3 Dependency Relations

A more general and perhaps more powerful way of converting dependency relations into syntactically relevant features is to include the specific dependency relations for each dependent of the target. For verbs, these include things like subjects, direct objects, adverbial modifiers, nominal modifiers, passive subjects, and more. Capturing the fine-grained dependencies for each verb is analogous to determining the exact syntactic construction it is being realized in. Combining this feature with the embeddings of dependents and heads is a promising avenue for linking syntax and semantics.

### 5.2.4 VerbNet Class

VerbNet is a lexical semantic resource that groups verbs into classes based on their syntactic behavior (Kipper-Schuler, 2005). It categorizes over 6,000 verbs into classes, each of which contains syntactic frames that the verbs in the class can appear in. It also contains distinct senses, allowing it to distinguish between different verb uses in context. Previous approaches have employed VerbNet as a lexical resource (Beigman Klebanov et al., 2016), but aggregated the senses of each verb, removing the syntactic distinctions that VerbNet makes for different word senses.

We ran word-sense disambiguation to determine the VerbNet class for each verb token (Palmer et al., 2017). We included one-hot vectors representing verb senses for each token, and combining this with knowledge of the particular constructions and the lexical semantics provided by embeddings for each token gives syntactically motivated information about the semantics of the utterance. For noun identification, we include the VerbNet class of the head of that noun.

## 5.3 Experiments

As a baseline, we began with using the embedding of the word to be classified. We concatenated this with the embeddings of the single previous and following words, as this proved the best context in our validation. This creates a representation of lexical semantics and a word’s context, without any specific knowledge of the syntactic relations the word is involved in. We then added each syntactic representation. These experiments were done using a training-validation-test split of 76/12/12. We experimented with Maximum Entropy, Naive Bayes, Random Forest and Support Vector Machine classifiers, and through validation chose a SVM with a linear kernel, L2 regularization and squared hinge loss. We then ran the classifier using our baseline, and added each feature separately. Finally, we combined the best feature

set for each classification task, judged by the improved performance of each feature over the baseline. The classification was split into three tasks: identifying source items, identifying target items, and identifying metaphoric (either source or target) from non-metaphoric. The results of these experiments are in table 2.

From these results we can see that classifying source-domain words in the LCC data is harder than classifying target-domain words. This may be because of the broad range of domains, as the corpus contains 114 possible source domains. Target items are much easier to classify, likely because the dataset contains only a limited number (32) of target domains. Embeddings are effective at representing semantics, and they can accurately determine the domain of lexical items, allowing for easy classification of target items.

Our syntactic features show mixed results. Adding sentential context is consistently effective, showing that naive contextual approaches are helpful. Adding dependency embeddings is also consistently effective, supporting our hypothesis that knowledge of syntactic properties can be helpful in metaphor classification. Other syntactic features are inconsistent, especially in predicting the metaphoricity of verbs. Selecting only the feature sets that showed improvement over the baseline yields the best results for most categories.

## 6 VUAMC Analysis

The LCC allows for an in-depth examination of source and target domains, but is relatively small compared to the VUAMC. We can use the VUAMC data to inspect the distribution of word metaphoricity with regard to argument structure constructions. While Sullivan’s work focuses on source and target domain elements and not whether or not words are used metaphorically, we can examine the binary classifications in the VUAMC to provide insight into the distribution of metaphoric verbs and the predicate-argument constructions they participate in. Counts of argument structure verbs and arguments and their metaphoricity are shown in table 3.

From the data in table 3, we can see clear distinctions between different constructions and the metaphoricity of their arguments. Verbs in intransitive constructions are much less likely to be metaphoric than those used in transitives, and both less so than those in ditransitive constructions.

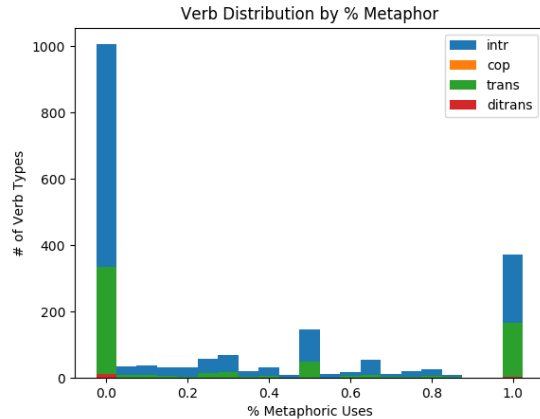


Figure 2: Verb types by percent of metaphoric use in each construction. Each bar represents the number of verb types that match the X axis for percentage of metaphoric usages.

The VUAMC chooses not to mark copular verbs as metaphoric, and only one instance was found of equative constructions having a metaphoric verb.

We might expect that different constructions would also impact the distribution of the predicates’ arguments. However, from the data we see that verb arguments are fairly consistent. Indirect objects in ditransitive constructions were never observed to be metaphoric, but direct objects are between 11% and 16% metaphoric throughout. Subjects vary from 2.8% in ditransitives to 11.7% in equative constructions. One distinctive feature is that subjects are much less likely than objects to be metaphoric.

The overall distribution of metaphoric uses by verb construction shows that the more arguments that are present in the construction, the more likely the verb is being used metaphorically. For further evidence, we can examine the distribution of metaphoric usages on a verb-specific basis.

We calculated the average metaphoricity of each verb found in the VUAMC, and sorted them by the type of construction they are found in. We performed this analysis on a type and token basis, shown in figures 2 and 3. From the data, we see that the majority of verbs in all constructions are used exclusively non metaphorically. While a large number of verb types only occur metaphorically, this accounts for a much smaller number of verb tokens. Verb types that occur only metaphorically are relatively rare. We can also see that ditransitive and copula verb types are exceedingly

Features	Verbs			Nouns		
	Src	Trg	Met	Src	Trg	Met
Baseline (Embedding, 1-word context)	.467	.316	.483	.440	.701	.597
+Context	<b>.494</b>	<b>.545</b>	.436	<b>.487</b>	<b>.705</b>	<b>.593</b>
+Dependent Embeddings	<b>.482</b>	<b>.421</b>	.444	<b>.570</b>	<b>.717</b>	<b>.631</b>
+Dependency Relations	<b>.488</b>	<b>.384</b>	.482	<b>.486</b>	<b>.718</b>	<b>.601</b>
+Argument Construction	.459	<b>.461</b>	.457	.456	.661	<b>.598</b>
+VerbNet Class	.467	<b>.555</b>	.473	.433	.684	.589
Best Combination	.551	.600	.505	.519	.705	.630

Table 2: Classification of Source and Target elements in the LCC Corpus. Metaphor (MET) is the classification of a word as either Source or Target against non-metaphoric words.

Verb	%	Predicate			Subject			Direct Object			Indirect Object		
		+M	-M	%Met	+M	-M	%Met	+M	-M	%Met	+M	-M	%Met
Intransitive	75.1	5118	24301	17.4	284	4627	5.8	-	-	-	-	-	-
Transitive	13.1	1517	3612	29.6	119	3125	3.7	654	4475	12.8	-	-	-
Ditransitive	.2	24	35	40.7	1	35	2.8	9	50	15.2	59	0	0
Equation	11.6	1	4548	.02	449	3376	11.7	468	3736	11.1	-	-	-
-Simile	.1	0	35	0.0	2	28	6.7	7	26	21.2	-	-	-

Table 3: % Metaphor by Construction (VUAMC). For each predicate, the count of metaphoric (+M) and non-metaphoric (-M) instances are counted, as well as those for all of each construction’s defining arguments.

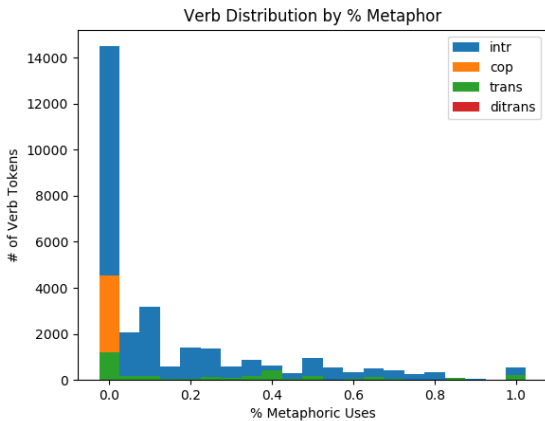


Figure 3: Verb tokens by percent of metaphoric use in each construction. Each bar represents the number of verb tokens that belong to verb types that match the X axis for percentage of metaphoric usages.

rare, but copula tokens are very common and almost always literal.

We extended this analysis by examining the distribution of the verb types that can appear intransitively, transitively, and ditransitively. Our hypothesis in studying these verbs is that the type of construction the verb appears in is predictive of that verb’s metaphoric use, independent of the verb’s overall behavior. Eleven verbs appeared in all three constructions, and the analysis of their

metaphoricity is presented in figure 4.

From the distribution in the VUAMC corpus, the data indicates that the type of argument structure construction does not significantly change the distribution of metaphoricity. The verbs generally have the same percentage of metaphoric usages regardless of which construction they appear in. Only ‘give’ appears in more than 2 instances of the ditransitive, and its distribution mirrors that of its use in other constructions.

Two components from our corpus analysis stand relevant for automatic metaphor processing. First, in broad scope over all verb tokens, predicates’ metaphor distributions are dependent on the kind of construction they occur in. Second, the verb itself is critical, as each verb tends to follow the same pattern of metaphoricity throughout its constructions. This supports our belief that identification of metaphor requires modeling of the interaction of syntactic and semantic information.

### 6.1 Metaphor Identification (VUAMC)

We employ the same experimental set up of the previous classification task using the VUAMC. The VUAMC doesn’t contain source or target annotations, so the classification problem is limited to identifying metaphoric words from non-metaphoric words. We employ the same baseline and syntactic representation features. Again, we

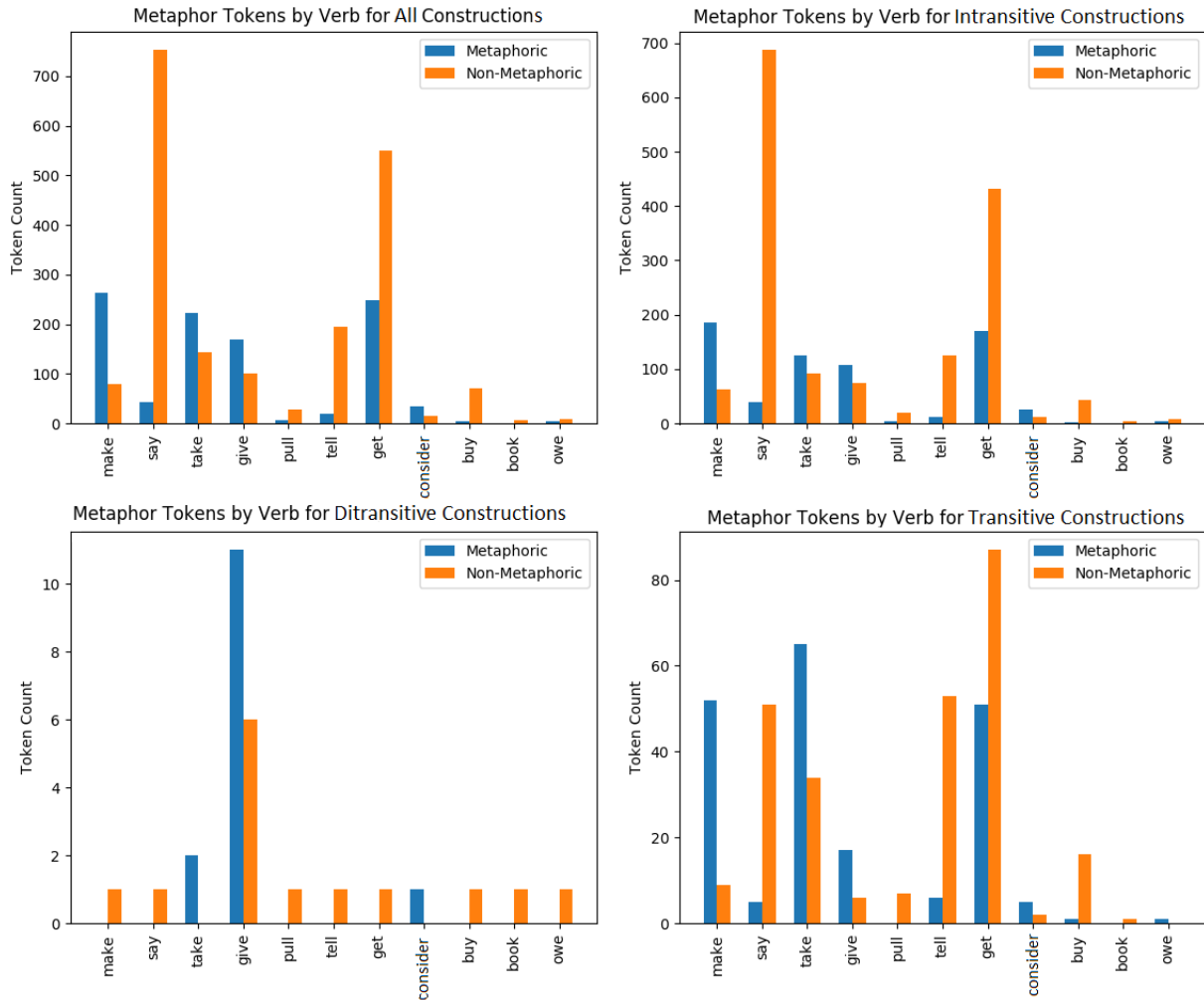


Figure 4: Counts of metaphoric uses by verb and construction for those verbs that occur in intransitive, transitive, and ditransitive constructions

used a split of 76/12/12, using a linear SVM.

For metaphoric identification in the VUAMC, all of the syntactic features improved classification over the baseline for verbs. For nouns, the dependency embeddings and VerbNet class of the noun’s head were effective. For both, combining all of the syntactic representations yields the best performance. While this classification based on syntactic is slightly lower than some recent experiments (Beigman Klebanov et al., 2016), it shows improvement over using purely lexical semantics, and we believe the incorporation of better syntactic representations can be used to improve metaphor identification systems.

## 7 Conclusions

The type of syntactic construction a verb is present in provides unique evidence of how it is being used metaphorically. It is important to effectively inte-

Model	Verbs	Nouns
Baseline (Embedding, 1-Word context)	.339	.303
+Context	<b>.488</b>	.224
+Dependency Embeddings	<b>.425</b>	<b>.349</b>
+Dependency Relations	<b>.466</b>	<b>.393</b>
+Argument Construction	<b>.471</b>	.289
+VerbNet Class	<b>.418</b>	<b>.330</b>
+All	.531	.505

Table 4: Results of adding different syntactic models for VUAMC verb classification.

grate syntax and semantics to detect and interpret metaphor, and because there are so many types of metaphors and they occur in such a wide array of contexts, it may be helpful to use separate methods of representing metaphoric semantics depending on the syntactic constructions involved. While our results indicate that these integrations of syntactic representations do not yet achieve state of the



art performance, we believe that improving representations of syntactic constructions can provide some benefit to metaphor processing.

To that end, our future goals include exploring better representations of the interaction between syntax and semantics. Models like syntactic tree kernels, compositional distributional semantic models, and other syntactically driven methods are likely to improve classification if they can properly combine syntactic and semantic representations. Additionally, as different constructions are likely to yield different types of metaphoricality, we aim to employ ensemble methods that incorporate construction-based knowledge to select the most effective classifier, and extending our approach to identifying source and target domains in addition to lexical triggers.

## Acknowledgements

We gratefully acknowledge the support of the Defense Threat Reduction Agency, HDTRA1-16-1-0002/Project #1553695, eTASC - Empirical Evidence for a Theoretical Approach to Semantic Components and a grant from the Defense Advanced Research Projects Agency 15-18-CwC-FP-032 Communicating with Computers, a sub-contract from UIUC. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any government agency.

## References

- Beata Beigman Klebanov, Chee Wee Leong, E. Dario Gutierrez, Ekaterina Shutova, and Michael Flor. 2016. [Semantic classifications for detection of verb metaphors](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 101–106, Berlin, Germany. Association for Computational Linguistics.
- Luana Bulat, Stephen Clark, and Ekaterina Shutova. 2017. [Modelling metaphor with attribute-based semantics](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 523–528, Valencia, Spain. Association for Computational Linguistics.
- Ellen Dodge, Jisup Hong, and Elise Stickles. 2015. [Metanet: Deep semantic automatic metaphor analysis](#). In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 40–49, Denver, Colorado. Association for Computational Linguistics.
- Andrew Gargett and John Barnden. 2015. Modeling the interaction between sensory and affective meanings for detecting metaphor. In *Third Workshop on Metaphor in NLP*, pages 21–30, Denver, CO.
- Jonathan Gordon, Jerry Hobbs, Jonathan May, and Fabrizio Morbini. 2015. [High-precision abductive mapping of multilingual metaphors](#). In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 50–55, Denver, Colorado. Association for Computational Linguistics.
- E. Dario Gutierrez, Ekaterina Shutova, Tyler Marghetis, and Benjamin Bergen. 2016. [Literal and metaphorical senses in compositional distributional semantic models](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 183–193, Berlin, Germany. Association for Computational Linguistics.
- Hessel Haagsma and Johannes Bjerva. 2016. [Detecting novel metaphor using selectional preference information](#). In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 10–17, San Diego, California. Association for Computational Linguistics.
- Ilana Heintz, Ryan Gabbard, Mahesh Srivastava, Dave Barner, Donald Black, Majorie Friedman, and Ralph Weischedel. 2013. [Automatic extraction of linguistic metaphors with lda topic modeling](#). In *Proceedings of the First Workshop on Metaphor in NLP*, pages 58–66, Atlanta, Georgia. Association for Computational Linguistics.
- Dirk Hovy, Shashank Shrivastava, Sujay Kumar Jauhar, Mrinmaya Sachan, Kartik Goyal, Huiying Li, Whitney Sanders, and Eduard Hovy. 2013. [Identifying metaphorical word use with tree kernels](#). In *Proceedings of the First Workshop on Metaphor in NLP*, pages 52–57, Atlanta, Georgia. Association for Computational Linguistics.
- Karen Kipper-Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania.
- George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago and London.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. [The Stanford CoreNLP natural language processing toolkit](#). In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.
- James H. Martin. 1990. *A Computational Model of Metaphor Interpretation*. Academic Press, Inc.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Distributed representations of words and phrases and their compositionality](#). *CoRR*, abs/1310.4546.

- Michael Mohler, Mary Brunson, Bryan Rink, and Marc Tomlinson. 2016. Introducing the lcc metaphor datasets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Ekaterina Ovchinnikova, Ross Israel, Suzanne Wertheim, Vladimir Zaytsev, Niloofar Montazeri, and Jerry Hobbs. 2014. [Abductive inference for interpretation of metaphors](#). In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 33–41, Baltimore, MD. Association for Computational Linguistics.
- Martha Palmer, James Gung, Claire Bonial, Jinho Choi, Orin Hargraves, Derek Palmer, and Kevin Stowe. 2017. The pitfalls of shortcuts: Tales from the word sense tagging trenches. *Essays in Lexical Semantics and Computational Lexicography - In Honor of Adam Kilgarriff*.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. [Glove: Global vectors for word representation](#). In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Pragglejaz Group. 2007. MIP: A method for identifying metaphorically used words in discourse. *Metaphor and Symbol*, 22(1):1–39.
- Sunny Rai, Shampa Chakraverty, and Devendra K. Tayal. 2016. [Supervised metaphor detection using conditional random fields](#). In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 18–27, San Diego, California. Association for Computational Linguistics.
- Marek Rei, Luana Bulat, Douwe Kiela, and Ekaterina Shutova. 2017. [Grasping the finer point: A supervised similarity network for metaphor detection](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1537–1546, Copenhagen, Denmark. Association for Computational Linguistics.
- Ekaterina Shutova. 2010. [Automatic metaphor interpretation as a paraphrasing task](#). In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1029–1037, Los Angeles, California. Association for Computational Linguistics.
- Ekaterina Shutova. 2013. [Metaphor identification and interpretation](#). In *Second Joint Conference on Lexical and Computational Semantics (\*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 276–285, Atlanta, Georgia, USA. Association for Computational Linguistics.
- Karen Sullivan. 2013. *Frames and Constructions in Metaphoric Language*. John Benjamins.
- Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. [Metaphor detection with cross-lingual model transfer](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258, Baltimore, Maryland. Association for Computational Linguistics.