Learning to Jointly Predict Ellipsis and Comparison Structures

Omid Bakhshandeh[1], **Alexis Wellwood**[2], **James Allen**[1,3]
1 University of Rochester, 2 Northwestern University, 3 Florida Institute for Human and Machine Cognition
{omidb, james}@cs.rochester.edu, wellwood@northwestern.edu

Abstract

Domain-independent meaning representation of text has received a renewed interest in the NLP community. Comparison plays a crucial role in shaping objective and subjective opinion and measurement in natural language, and is often expressed in complex constructions including ellipsis. In this paper, we introduce a novel framework for jointly capturing the semantic structure of comparison and ellipsis constructions. Our framework models ellipsis and comparison as interconnected predicate-argument structures, which enables automatic ellipsis resolution. We show that a structured prediction model trained on our dataset of 2,800 gold annotated review sentences yields promising results. Together with this paper we release the dataset and an annotation tool which enables two-stage expert annotation on top of tree structures.

1 Introduction

Representing the underlying meaning of text has been a long-standing topic of interest in computational linguistics. Recently there has been a renewed interest in representation of meaning for various tasks such as semantic parsing, where the task is to map a natural language sentence into its corresponding formal meaning representation (Zelle and Mooney, 1996; Berant and Liang, 2014). Open-domain and broad-coverage semantic representation of text (Banarescu et al., 2013; Bos, 2008; Allen et al., 2008) is crucial for many language understanding tasks such as reading comprehension tests and question answering. With the rise of continuous-space models there is even more interest in capturing deeper generic semantics of text as opposed to surface word representations.

One of the most common ways for expressing evaluative sentiment towards different entities is using comparison. A simple natural language example of comparison is *Their pizza is the best*. Capturing the underlying meaning of comparison structures, as opposed to their surface wording, is required for accurate evaluation of qualities and quantities. For instance, given a more complex comparison example, *The pizza was great, but it was not as awesome as the sandwich*, the state-of-the-art sentiment analysis system (Manning et al., 2014) assigns an overall 'neutral' sentiment value, which clearly lacks deeper understanding of the comparison happening in the sentence.

Consider the generic meaning representation depicted in in Figure 1 according to frame semantic parsing [1] (Das et al., 2014) for the following sentence:

(1) My Mazda drove faster than his Hyundai.

It is evident that this meaning representation does not fully capture how the semantics of the adjective *fast* relates to the *driving* event, and what it actually means for a car to drive *faster than* another car. More importantly, there is an ellipsis in this sentence, the resolution of which results in complete understood reading of *My Mazda drove faster than his Hyundai* $_{drove\ fast}$, which is in no way captured in Figure 1[2]. Having a comprehensive meaning representation of comparison struc-

[1] We used Semafor tool: http://demo.ark.cs.cmu.edu/parse

[2] The same shortcomings are shared among other generic meaning representations such as LinGO English Resource Grammar (ERG) (Flickinger, 2011), Boxer (Bos, 2008), or AMR (Banarescu et al., 2013), among others.

Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL), pages 62–74,
Berlin, Germany, August 7-12, 2016.

My Mazda	drove	faster than his Hyundai
Self_mover	**Self_motion**	Manner

Figure 1: The frame-semantic parsing of the sentence *My Mazda drove faster than his Hyundai.*

tures which can capture the mentioned phenomena can enable the development of computational semantic models which are suitable for various reasoning tasks.

In this paper we introduce a joint theoretical model for comprehensive semantic representation of the structure of comparison and ellipsis in natural language. We jointly model comparison and ellipsis as inter-connected predicate-argument structures, which enables automatic ellipsis resolution. The main contributions of this paper can be summarized as follows: (1) introducing a novel framework for jointly representing the semantics of comparison and ellipsis on top of syntactic trees, (2) releasing a dataset of 2,800 expert annotated user review comparison instances[3], which significantly increases the size of the available resources on comparison structures in the community, (3) presenting a new structured prediction model for automatic extraction of semantic structures of comparison text together with ellipsis resolution, (4) releasing an interactive tool for tree-based human annotation of corpora, which can be helpful for many other annotation tasks in NLP.

To our knowledge, this paper presents the first comprehensive computational framework of its kind for ellipsis and comparison constructions. Our semantic model can be incorporated as a part of any broad-coverage semantic parser (Banarescu et al., 2013; Allen et al., 2008; Bos, 2008) for augmenting their meaning representation.

2 Background and Related Work

Broadly, elliptical constructions involve the omission of one or more phrases from a clause (such as 'drove fast' phrase at the end of example (1)) whose content can still be fully recovered from the unelided words of the sentence (Kennedy, 2003; Merchant, 2013). Resolving ellipsis is crucial for deep language understanding. Although ellipsis has been studied in great depth in linguistics, there only have been a few computational studies of ellipsis, most of which have focused on Verb Phrase Ellipsis (VPE) (Nielsen, 2004; Schiehlen, 2002; Hardt, 1997) such as *Larry is not telling the truth, neither is Jim* Δ. where Δ is a verb phrase ellipsis site, which can be resolved to 'telling the truth'.

In 2010, a SemEval task was organized with the goals of (1) automatically detecting VPE in text, and (2) resolving the antecedent of each VPE (Bos and Spenader, 2011). For this task, they manually annotated a portion of OntoNotes corpus, consisting of Wall Street Journal (WSJ) articles. Throughout all the 25 sections of WSJ, they found 487 instances of VPE (ranging from predicative ellipsis, deletion, and comparative constructions, to pseudo-gapping) in about 53,600 sentences. Among 487 ellipsis items, there were 96 comparative constructions. They show that simply searching the parse trees for empty VPs achieves a high precision (0.95) but low recall (0.58). Our work presents the first attempt on comparison ellipsis resolution of various types, within a semantically rich framework of comparisons.

The syntax and semantics of comparison structures in natural language have been the subject of extensive systematic research in linguistics for a long time (Bresnan, 1973; Cresswell, 1976; Von Stechow, 1984). Measurement in language is mainly expressed by using comparative morphemes such as *more, less, -er, as, too, enough, -est, etc*[4]. The main component of the sentence carrying out the measurement can have either of adjective (JJ), adverb (RB), noun (NN), or verb (VB) parts of speech. The earliest efforts on the computational modeling of comparatives have been in the context of sentiment analysis, ranging from works on identifying sentences containing comparisons (Jindal and Liu, 2006b) to identifying the components of the comparisons in the form of triplets or other templatic patterns (Jindal and Liu, 2006a; Xu et al., 2011; Kessler and Kuhn, 2014). These works provide a basis for computational analysis of comparatives, however, they lack depth and broader coverage as they are limited to only a few comparison patterns.

The most recent work on the computational semantics of comparison (Bakhshandeh and Allen, 2015) sets the stage for a deeper semantic representation of comparisons. Bakshandeh and Allen introduce the first computational semantic frame-

[3]Throughout this paper we refer to any statement comparing two or more entities as a comparison instance.

[4]These morphemes are often referred to as the comparison operators.

work for representing the meaning of comparatives in natural language. This framework models comparisons as predicate-argument pairs interconnected via semantic role links. Our framework differs in the following crucial aspects:

– **Joint Ellipsis and Comparison Modeling**: Effective modeling and reasoning on comparison structures requires addressing ellipsis as well. While Bakhshandeh and Allen only model comparisons, we provide a novel semantic framework for comprehensive annotation of ellipsis structures within comparison structures (details in Section 3.2).

– **Tree-based Structure Modeling**: Bakhshandeh and Allen use span-based predicate-argument treatment, which is often prone to errors and lower inter-annotator agreement. We base our framework on top of constituency syntactic parse trees, which leads to more accurate[5] capture of semantic structures.

– **Reviews Dataset**: While Bakhshandeh and Allen use newswire text, we shift our focus to the actual user reviews, which contain more comparison structures (Section 4.2). Furthermore, while their dataset included 531 sentences, we collect gold annotations for 2,800 sentences, which significantly increases the size of the available data for the community.

3 A Comprehensive Semantic Framework for Comparison

In this Section we introduce a novel semantic framework of comparison structures which incorporates ellipsis. Our framework extends and improves the state-of-the-art semantic framework for comparison structures in various ways (outlined in Section 2). We follow the model of interconnected predicate-argument structures. In this model the predicates are either comparison or ellipsis operators, and each predicate takes a set of arguments called its *semantic frame*. For instance, in *[Sam] is the tallest [student] [in the gym]*, the morpheme *-est* expresses a comparison operator and the brackets delimit its various arguments. In this Section we provide details about our semantic framework.

[5]This is crucial, given the fact that the syntactic structure of many comparison instances are complex, e.g., *The server was the rudest ever and made me feel as I was wasting her time.*

3.1 Comparison Structures

Comparison structures are modeled as sets of inter-connected predicate-arguments. We base our comparison framework on Bakhshandeh and Allen (Bakhshandeh and Allen, 2015), however, we extend and improve on the set of predicate types and arguments to capture more diverse structures which results in a different semantic framework.

3.1.1 Predicates

We consider two main categories of comparison predicates, each of which can grade any of the four parts of speech including adjectives, adverbs, nouns, and verbs.

1. **Ordering**: Indicates how two or more entities are ordered along a scale. The subtypes of this predicate are the following:

– Comparatives with '$>$', '$<$' indicate that one degree is greater or lesser than another; expressed by the morphemes *more*/*-er* and *less*.

(2) The steak is tastier than the potatoes.

(3) Tom ate more soup.

– Equatives involving '$\geq$' indicate that one degree meets or exceeds another; expressed by *as* in constructions such as *as tall* or *as much*.

(4) The Mazda drives as fast as the Nissan.

– Superlatives indicate an entity or event has the 'highest' or 'lowest' degree on a scale; expressed by *most*/*-est* and *least*.

(5) That chef made the best soup.

2. **Extreme**: Indicates having too much or enough of a quality or quantity. The subtypes of this predicate are the following:

– Excessive indicate that an entity or event is 'too high' on a scale; expressed by *too*.

– Assetive indicate that an entity or event has 'enough' of a degree; expressed by *enough*.

3.1.2 Arguments

Each predicate takes a set of arguments that we refer to as the predicate's 'semantic frame'. Following are the arguments included in our framework:

– Figure (Fig) is the main role which is being compared.

– Ground is the main role Figure is compared to.

– Scale (-/neutral/+) is the scale for the comparison, such as size, beauty, temperature. We assign the generic sentiment values positive (+), neutral, and negative (-) to the underlying scales.

– Standard (Std) is the reason a degree is 'too much' (excessive predicates) or 'enough' (assetive predicates). An individual j may be 'too tall *to reach the top shelf*' but 'tall enough *to get on this ride*'.

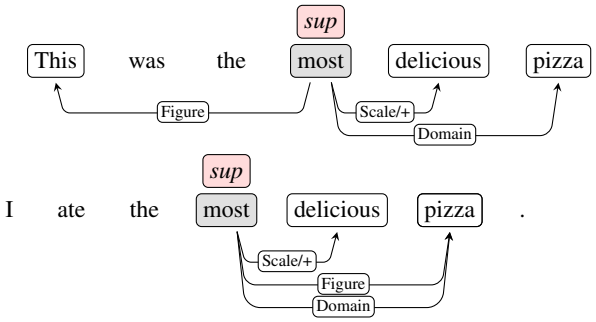
– Differential (Diff) is an explicit phrase indicating the 'size' of a difference between degrees. For instance, '*2 inches* taller' or '*6 degrees* warmer'.

– Domain (Dom) is an explicit expression of the type of domain in which the comparison takes place (superlatives). An individual m may be 'the tallest *girl*' but not 'the tallest *student*'.

– Domain Specifier (D-Spec) is the specification of the domain argument, further narrowing the scope of the domain. An individual m may be 'the tallest girl *in the class*' but not 'the tallest girl *in the country*'.

The Case of Copulas: A copula is a form of the verb *to be* that links the subject of a sentence with a predicate, such as *was* in the sentence *She was a doctor*. Comparison structures are often formed on the basis of copular constructions, for example (6a). Compare this with (6b), and their corresponding comparison structures.

(6) a. This was the best pizza in town.
b. I ate the best pizza in town.



As you can see, in (6a) *was* links *this* to *pizza*. In this sentence the argument Figure is *this*. On the other hand, in (6b), the word *pizza* takes the role of both Figure and Domain.

3.2 Ellipsis Structures

Perhaps the most common type of comparison structure is the comparative construction, with (13) as an example, where Δ marks an ellipsis site. Roughly, (13) is interpreted as a greater-than relation between 'how appetizingly the steak sizzles' and 'how appetizingly the hamburger sizzles', which might be formalized as in (14) with e_1 and e_2 representing the two sizzling events.

(7) The steak sizzled more appetizingly than the hamburger Δ.

(8) $appetizingness(e_1) > appetizingness(e_2)$

On the surface, the sentence in (13) does not relate *sizzle* or *appetizingly* to the hamburger; these must be filled in for Δ by a process called *ellipsis resolution*—finding the *antecedent* of an ellipsis. Speakers of English are readily able to infer from the surface material in (13) that the dependent clause is interpreted as in (9), where the resolved ellipsis is written in subscript.

(9) than the hamburger$_{\text{sizzled appetizingly}}$

It is clear that resolving ellipsis in comparison structures is crucial for language understanding and failure to do so would deliver an incorrect meaning representation. Numerous subtypes of elliptical constructions are distinguished in linguistics (Kennedy, 2003; Merchant, 2013; Yoshida et al., 2016). In our framework we mainly include six types that can be detected in comparison structures: 'VP-deletion', 'Stripping'[6], 'Pseudogapping', 'Gapping', 'Sluicing', and 'Subdeletion'. Ellipsis more often occurs in comparative and equative constructions (applicable to any of the four parts of speech) as follows.

• **Comparatives**: Ellipsis takes place in the dependent clause headed by *than*. We indicate the three ellipsis possibilities for these clauses resuming (10), a nominal comparative. The elided materials are written in subscript.

(10) **Mary ate more rice** ...

– VP-deletion (aka 'Comparative Deletion'):
... than John did $_{\text{eat rice}}$.
– Stripping (aka 'Phrasal Comparative'):
... than John $_{\text{ate rice}}$.
– Gapping:
... than John, $_{\text{ate } \textit{how-much}}$ soup.
– Pseudogapping:
... than John did $_{\text{eat}}$ soup.
– Sluicing:
... than someone, but I don't remember than who $_{\text{ate how-much rice}}$.
– Subdeletion:
... than John ate $_{\textit{how-much}}$ soup.

• **Equatives**: Ellipsis takes place in the dependent clause headed by *as*. We indicate the possibilities for these clauses resuming (11), a nominal equative.

(11) **Mary ate as much rice** ...

– VP-deletion:
... as John did $_{\text{eat } \textit{how-much} \text{ rice}}$.

[6] VP-deletion and stripping are the more frequent types.

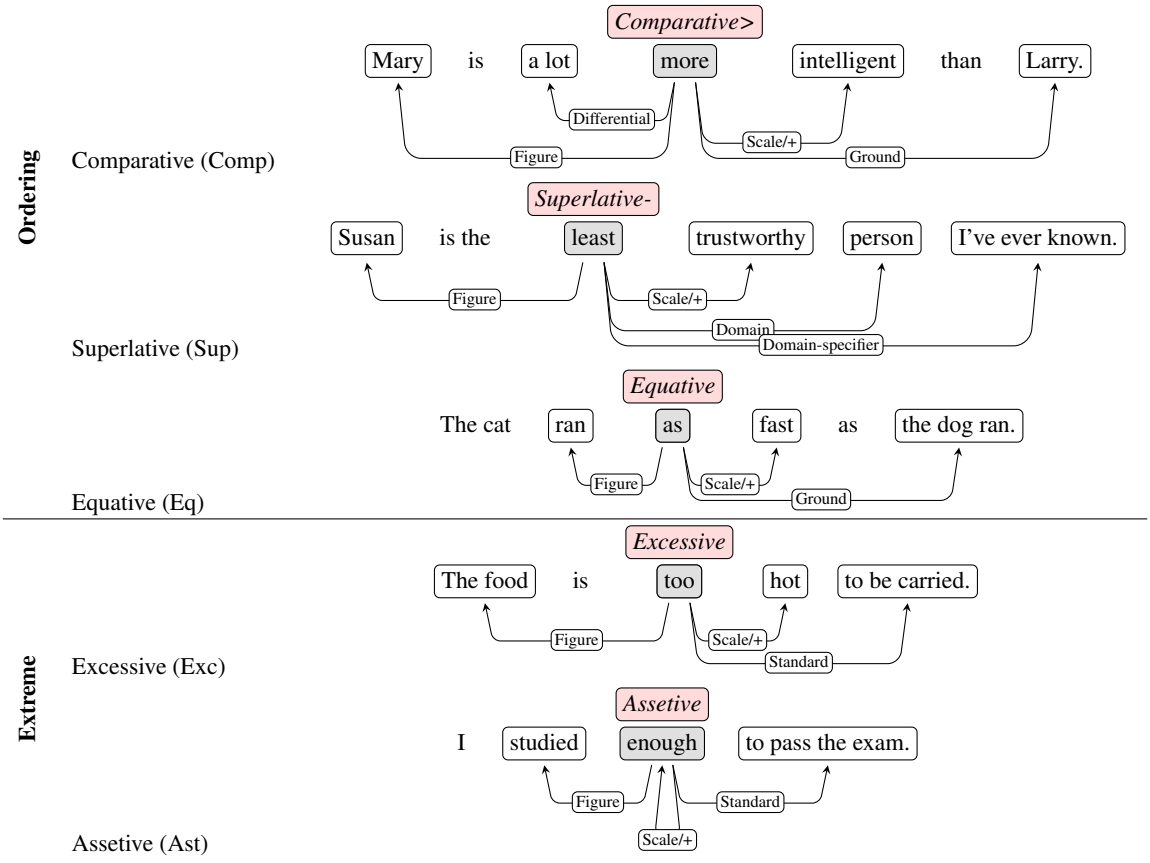


Table 1: Predicates together with their semantic frames shown in example sentences.

– Stripping:
  ... as John eat *how-much* rice.
– Gapping:
  ... as John, ate *how-much* soup.
– Pseudogapping:
  ... as John did ate *how-much* soup.
– Sluicing:
  ... as someone, but I don't remember as who ate how-much rice.[7]
– Subdeletion:
  ... as John ate *how-much* soup.

Now that we have the ellipsis predicate types, we want to empirically model ellipsis constructions as predicate-argument structures with reference to an antecedent, where each ellipsis predicate is associated with its corresponding comparative predicate. The question is how to represent the ellipsis construction in a sentence. Consider the example of VP-deletion in the following adverbial comparative:

(12) The steak was cooked more carefully than the burger Δ.

where Δ should be resolved to *was cooked how-carefully*. *How* is called the null operator, which serves as the placeholder for the measurement of a degree.

In order to represent the resolution of the elided material such as Δ, we first annotate the predicate of an ellipsis construction as an 'attachment' site in the syntactic tree, right next to the node that the elided material should follow. Hence, in (12), the token *the burger* will be annotated as the ellipsis predicate, which signifies the start of an ellipsis construction.

Defining the arguments for ellipsis predicates can be complicated. Here the goal is to thoroughly construct the antecedent of the elided material by annotating the existing words of the context sentence. In order to address this, we define the following three argument types for ellipsis:

– Reference is the constituency node which is the base of an antecedent.
– EXclude (Ex) is the constituency node which should be excluded from the Reference.
– How-much (?) is the constituency node which should be replaced by a null operator such as *how* or *how-much*; this is always the argument matching *more*/*-er* or *as (much)* in the context sentence.

[7] Whether this construction is grammatical is controversial.

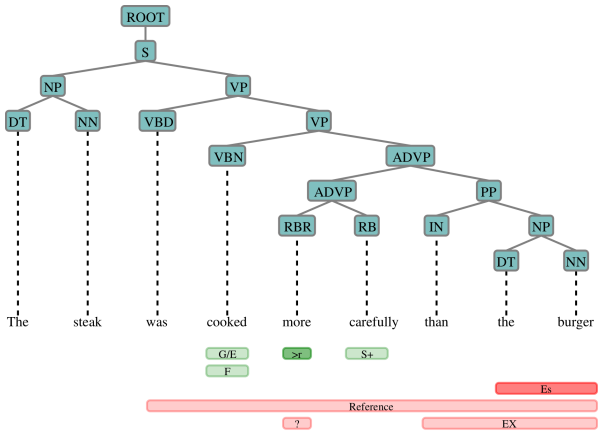


Figure 2: Full tree-based annotation of comparison and ellipsis structures for the sentence presented in example 12. The tag 'Es' refers to the Stripping predicate type.

Following the above annotation schema the ellipsis site in (12) will be annotated as shown in red in Figure 2. This shows how to do automatic ellipsis resolution given our representation: one should start *after* the node 'the burger', and perform the following: [was cooked ~~more~~$_{How?}$ carefully than the burger]$_{Reference}$ − [than the burger]$_{EXclude}$ = *was cooked how carefully.* Another important thing to note in Figure 2 is our treatment of the comparison structure (in green) jointly with ellipsis: The argument F (Figure) of the comparison predicate *more* is *cooked*. The G argument (Ground), is the second elided 'cooked' event, which should come from the ellipsis construction. We thus annotate the explicit node *cooked* as the Ground-Ellipsis (G/E) which also links the comparison construction to the ellipsis predicate.

4 Data Collection Methodology

4.1 Comparison Instance Sampling

The sentences used for annotation play a significant role in the diversity and comprehensiveness of the comparison structures represented in our dataset. Earlier work (Bakhshandeh and Allen, 2015) experimented with annotating semantic structures on OntoNotes dataset. We shift our focus to actual product and restaurant reviews, which inherently include many natural comparison instances. For this purpose we mainly use Google English Web Treebank[8]. This corpus contains more than 250,000 words in about 10,000 sentences of English weblogs, newsgroups, email, reviews (product, restaurant, etc.) and question-answers, annotated with gold syntactic trees. This corpus is suitable for our task since it provides a good coverage of web domain text, mainly reviews.

In order to augment the volume of review content, we also use the Movie Reviews dataset (Pang and Lee, 2005). This dataset consists of 11,855 sentences extracted from movie reviews. Given that these Movie reviews do not come with the syntactic trees, we used the Berkeley parser (Petrov et al., 2006), which outperformed the other off-the-shelf parsers on comparison syntactic structure. Of course it is not efficient to include any arbitrary sentence of a corpus for manual annotation. We employ various linguistic filters to filter the sentences which potentially contain comparison. The details of this process can be found in the supplementary material.

4.2 Tree-based Annotation

We trained six linguists to do the semantic annotation for comparison and ellipsis structures for the sampled comparison instances according to the framework presented in Section 3. The annotations were done via our interactive two-stage tree-based annotation tool. In this tool, each annotator can be assigned with a set of tree-based annotation assignments, where pairing annotators to do the same task for inter-annotator analysis is also feasible. For this task, the annotations were done on top of constituency parse trees, and the annotators were instructed to choose the top-most constituency node when choosing the predicate or arguments.[9] Annotating on gold-standard syntactic trees helps with resolving ambiguous instances which have multiple interpretations. Furthermore, it gives annotators syntactic signals for choosing the types of predicates (e.g., adverbial vs adjectival comparatives), all of which increase the accuracy of our annotation.

Our annotation tool sets up the data collection as a two-stage expert annotation process: (1) for each sentence, one expert annotates and submits the annotation, (2) another expert reviews the submission and either returns the submission with feedback or marks it as a gold. This recursive

[8] https://catalog.ldc.upenn.edu/LDC2012T13

[9] This enables accurate capturing of arguments, e.g., in *I am the tallest [in our school]*, the constituency node corresponding to the entire phrase in brackets is annotated as Domain-specifier.

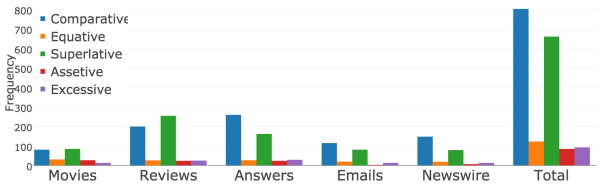


Figure 3: The number of various predicate types across different resources.

Table 2: The percentage of each argument type.

Scale	Fig	Ground	Dom	D-Spec	Diff	Std
38.8	31.5	6.33	9.31	7.01	4.09	2.98

process ensures higher annotation quality. We iterate over the sentences until getting 100% inter-annotator agreement. On average, annotating every sentence takes about one minute and revising controversial sentences (12% of the time) takes about 4 minutes of expert annotation time.

This process yields a total of 2,800 annotated sentences with 100% agreement. Figure 3 visualizes the distribution of various predicate types from the various resources. In order, these resources each include 11,855, 3,813, 3,488, 4,900, and 2,391 sentences. As this Figure depicts, reviews are indeed the richest resource for comparisons, with more comparison instances than any other resource of even a bigger size. There are a total of 5,564 comparison arguments in our dataset, with the distribution summarized in Table 2. The total number of ellipsis predicates is 240, with 197 Stripping, 31 VP-deletion and 12 Pseudo-gapping.

5 Predicting Semantic Structures

In this Section we describe our methodology for joint prediction of comparison and ellipsis structure for a given sentence.

5.1 Modeling

We model the problem as a joint predicate-argument prediction of comparison and ellipsis structures. It is important to note that our predicate-argument semantic structure itself looks similar to a dependency parse tree, however, as explained earlier, we base this representation on top of constituency parse trees. For each training sentence, we denote the underlying constituency tree as T. The set of all constituency nodes in T is V_T. Each $v \in V_T$ can be tagged as a comparison predicate $c \in C$ = *{Comp, Sup, Eq, Exc, Ast}*[10], a comparison argument $a_c \in A_C$ = *all-comparison-arguments*, an ellipsis predicate e = 'Ellipsis', an ellipsis argument $a_e \in A_E$ = *{Reference, Ex, '?'}*, or *NONE*.

In Equation 1, we define a globally normalized model for the probability distribution of comparison labels over all $v \in V_T$ if $CompFilter(T)$ = True. We define $CompFilter$ to filter the following:

– Any sentence containing a word with POS tag equal to *JJR*, *RBR*, *JJS*, or *RBS*.

– Any sentence containing a comparison morpheme such as *more*, *most*, *less*, *enough*, *too*.

The next step is to define the probability distribution in Equation 2 for ellipsis labels, conditioning on the comparison label. This is motivated by the fact that the Ellipsis predicate is dependent on its corresponding comparison predicate. Given the comparison and ellipsis predicate labels, for each comparison and ellipsis argument type we define a binomial probability distribution as defined in Equations 3 and 4.

$$p_C(c|v,T,\theta_C) \propto exp(\boldsymbol{f}_C(c,T)^T \boldsymbol{\theta_C}) \quad (1)$$

$$p_E(e|c,v,T,\theta_E) \propto exp(\boldsymbol{f}_E(e,c,T)^T \boldsymbol{\theta_E}) \quad (2)$$

$$p_{A_c}(a_c|c,e,v,T,\theta_{a_c}) \propto exp(\boldsymbol{f}_{A_C}(c,e,T)^T \boldsymbol{\theta_{a_c}}) \quad (3)$$

$$p_{A_e}(a_e|c,e,v,T,\theta_{a_e}) \propto exp(\boldsymbol{f}_{A_E}(e,c,T)^T \boldsymbol{\theta_{a_e}}) \quad (4)$$

In each of the above equations, f is the corresponding feature function. For predicates the main features are lexical features, bigram features, node's constituency position, node's minimum distance from leaves, and node's parent constituency label. For the arguments, we use the same feature-set as for the predicates, but also including the leftmost verb (for the case of copulas), the constituency path between argument and the predicate, and the predicate type. θ_C, θ_E, θ_{a_c} and θ_{a_e} are the parameters of the log-linear model. We calculate these parameters using Stochastic Gradient Descent algorithm.

5.2 Joint Inference of Ellipsis and Comparison

For inference we model the problem as a structured prediction task. Given the syntactic tree of a given sentence, for each node we first select the predicate type with the highest p_C. Then for each

[10] *Each predicate should be further tagged with one of the four possible POS tags (JJ, RB, NN, VB), resulting in a total of* 20 *predicate types.*

selected comparison predicate, we find the corresponding ellipsis predicate that has the highest p_E probability. Define $\langle tc, te \rangle \in R$, where R is the set of all tuples of corresponding comparison and ellipsis predicates, tc is the index of the comparison predicate and te is the index of the ellipsis predicate.

We tackle the problem of argument assignment by using Integer Linear Programming, where one can pose domain-specific knowledge as constraints. We define a binary variable b_{ij} and b'_{ik} where i is the a node in tree, j is a comparison argument label and k is a ellipsis argument label. For each $\langle tc, te \rangle$, we maximize the linear Equation 5, subject to a few linguistically-motivated constraints.

$$\max_{b_{ij}, b'_{jk} \in \{0,1\}} \sum_{i \in V_T, j \in A_C, k \in A_E} \Big(b_{ij} p_{A_c}(tc, te, i, j) + b'_{ik} p_{A_e}(tc, te, i, k) \Big) \quad (5)$$

ILP Constraints: Any specific comparison label calls for a unique set of constraints in the ILP formulation, which ensures the validity of predictions. For instance, the $Superlative$ predicate type never takes any $Ground$ arguments, or the argument $Standard$ is only applicable to the excessive predicate type. We implement the semantic frame (as listed in Table 1) of each predicate type using hard ILP constraints. For example, in order to encode the semantic frame for predicate type $Excessive$, we employ the ILP constraints in Equation 6, which simply enforce this predicate to have 0 Ground arguments and maximum 1 Figure arguments.

$$\sum_{i \in V_T, j = Ground} b_{ij} = 0, \quad \sum_{i \in V_T, j = Figure} b_{ij} \leq 1 \quad (6)$$

We incorporate a few other ILP constraints for encoding our knowledge regarding ellipsis structures as well as comparison. For more details of these knowledge-driven constraints please refer to the supplementary material.

6 Experimental Result

We divide our dataset into train and train-dev (70%), and test (30%) sets. For evaluation of a given system prediction against the reference gold annotation, for each constituency node in the reference, we give the system a point in two ways:

	ILP Model		
	P	**R**	**F1**
Excessive	0.68/0.68	1.00/1.00	0.81/0.81
Assetive	0.97/0.97	1.00/1.00	0.98/0.98
Comparative	0.95/0.95	0.99/0.99	0.97/0.97
Superlative	0.97/0.98	0.98/0.99	0.98/0.98
Equative	0.57/0.58	0.95/0.98	0.71/0.73
Stripping	0.75/0.96	0.75/0.96	0.75/0.96
Deletion	0.20/0.41	0.72/0.89	0.31/0.13
Average	0.72/0.78	0.91/0.97	0.76/0.80
		Baseline	
Excessive	0.65/0.65	1.00/1.00	0.79/0.79
Assetive	0.97/0.97	1.00/1.00	0.98/0.98
Comparative	0.95/0.97	0.96/0.97	0.95/0.97
Superlative	0.98/0.98	0.98/0.98	0.98/0.98
Equative	0.13/0.13	1.00/1.00	0.23/0.23
Stripping	0.05/0.14	0.31/0.91	0.08/0.25
Deletion	0.00/0.00	0.00/0.00	0.00/0.00
Average	0.62/0.64	0.87/0.97	0.66/0.69

Table 3: Predicate prediction results on test set. Each cell contains scores according to Exact/Head measurement.

(1) Exact: the label assigned to the node by the system exactly matches the gold label; (2) Head: the reference label matches the label of the head word of the node in system's prediction. We report on Precision (P), Recall (R) and F1 score. We test three models: our comprehensive ILP model (detailed in Section 5), our model without the ILP constraints, and a rule-based baseline. The baseline encodes the same linguistically motivated ILP constraints via rules. It further uses a few pattern extraction functions for pinpointing comparison morphemes which detect comparison and ellipsis predicates. More details about the baseline can be found in the supplementary material.

The results on predicate prediction is shown in Table 3. Given that our ILP constraints only encode argument structures, in this Table we only compare the baseline with our full ILP model. As the results show, overall, the scores are high for predicting the predicates, with ellipsis predicates being the most challenging. The baseline has a near perfect prediction on Assetive and Superlative types, which shows that the linguistic patterns can capture these types well. Our model performs the poorest on $Equatives$. If we look at the specific cases it misses, it is often regarding the morpheme 'as', which takes part in many various linguistics constructions, many of which are not comparatives. For example, for the test sentence *We will let them manage our other investment properties as well as us getting older.*, our system wrongly classifies 'as' as an equative

	ILP Model (Exact/Head)			**ILP No Constraints** (Exact/Head)			**Baseline** (Exact/Head)		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Standard	0.40/0.80	0.42/0.84	0.41/0.82	0.00/0.00	0.71/1.00	0.00/0.00	0.00/0.00	0.00/0.00	0.00/0.00
Scale	0.58/0.64	0.89/0.99	0.70/0.78	0.02/0.02	0.94/1.00	0.04/0.04	0.47/0.69	0.67/0.98	0.55/0.81
Ground	0.27/0.48	0.46/0.84	0.34/0.61	0.00/0.00	0.98/1.00	0.01/0.01	0.06/0.18	0.24/0.71	0.10/0.29
Figure	0.38/0.81	0.44/0.94	0.41/0.87	0.02/0.02	0.94/1.00	0.03/0.03	0.09/0.43	0.17/0.80	0.12/0.56
D-Specifier	0.41/0.63	0.57/0.87	0.48/0.73	0.00/0.00	1.00/1.00	0.01/0.01	0.00/0.00	0.00/0.00	0.00/0.00
Domain	0.56/0.76	0.66/0.91	0.61/0.83	0.01/0.01	0.99/1.00	0.01/0.01	0.00/0.39	0.00/0.55	0.00/0.46
Exclude	0.33/0.56	0.49/0.84	0.39/0.67	0.01/0.01	0.63/1.00	0.02/0.02	0.00/0.00	0.00/0.00	0.00/0.00
Ref	0.18/0.53	0.28/0.80	0.22/0.63	0.01/0.01	0.61/1.00	0.01/0.02	0.00/0.00	0.00/0.00	0.00/0.00
How-much	0.27/0.36	0.65/0.88	0.38/0.51	0.01/0.01	0.96/1.00	0.01/0.01	0.00/0.00	0.00/0.00	0.00/0.00
Average	0.37/0.61	0.54/0.87	0.43/0.71	0.01/0.01	0.86/1.00	0.10/0.10	0.20/0.42	0.36/0.73	0.25/0.52

Table 4: Results of argument prediction on test set. The average for the models only takes into account non-zero results.

predicate, which is clearly an ambiguous and challenging test sentence. Analysis shows that the errors are often due to inaccuracies in automatically generated parse trees, e.g., challenging long sentences (average length > 12 tokens) with informal language which are generally hard to parse.

The task of predicting arguments is a more demanding task. As you can see in Table 4, the baseline model often fails at predicting the arguments. Our comprehensive ILP model consistently outperforms the *No Constraints* model, showing the effectiveness of our linguistically motivated ILP constraints. Our ILP model performs the best on Scale and Domain argument types, which is partly due to the frequency of these types in our dataset. We are planning on annotating more data to improve the argument prediction in future.

7 Conclusion

Systems that can understand comparison and make inferences about how entities and events compare in natural language are crucial for various NLP applications, ranging from question answering to product review analysis. Having a comprehensive semantic framework which can represent the underlying meaning of comparison structures is the first step toward enabling such an inference. In this paper we introduced a novel semantic framework for jointly capturing the meaning of comparison and ellipsis constructions. We modeled the problem as inter-connected predicate-argument prediction. Based on this framework, we trained experts to annotate a dataset of ellipsis and comparison structures, which we are making publicly available[11]. Furthermore, we introduced a structured prediction model which can automatically extract comparison structures and perform ellipsis resolution for a given text, which performs reasonably well for major predicate and argument types.

In future, we are planning on improving our joint prediction models for further improving the performance. Moreover, we plan on using our semantic framework for text comprehension applications.

Acknowledgment

We thank the anonymous reviewers for their invaluable comments and Brian Rinehart and other annotators for their great work on the annotations. This work was supported in part by Grant W911NF-15-1-0542 with the US Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO).

[11] In order to access the dataset and our interactive two-stage tree-based annotation tool please refer to `http://cs.rochester.edu/~omidb`.

Supplementary Material

Background on Ellipsis

Elliptical constructions involve the omission of one or more phrases from a clause, while the content can still be understood from the rest of the sentence (Kennedy, 2003; Merchant, 2013). Resolving ellipsis in comparison structures is crucial for language understanding. Failure to do so for (13) as an example, would deliver an incorrect representation, something like 'how appetizingly the steak sizzled is greater than the hamburger'. To arrive at an interpretation equivalent to (14) in a way that systematically relates to the syntax of (13) requires a semantics for comparatives based on 'events' and 'degrees'.

(13) The steak sizzled more appetizingly than the hamburger Δ.

(14) $appetizingness(e_1) > appetizingness(e_2)$

In event semantics, sentences like (15) and (16) are interpreted as existential statements about events (Davidson, 1967). For example, (15) is interpreted as 'there is an event e whose Theme (Th) is the steak, and e is a sizzling event' (Parsons, 1990).

(15) The steak sizzled. $\rightsquigarrow$
$\exists e_1[Th(e_1)(steak)$ & **sizzle**$(e_1)]$

(16) The hamburger sizzled. $\rightsquigarrow$
$\exists e_2[Th(e_2)(hamburger)$ & **sizzle**$(e_2)]$

A comparative like (13) is built on top of two clauses much like (15) and (16) (Bresnan, 1973). In concert with *appetizingly* in (13), *more* introduces a greater-than relation between the degrees to which the two events are appetizing (Wellwood, 2015). 'Degrees' represent points on a scale, said to be the output of a 'measure function' like **appetizing** (Cresswell, 1976; Kennedy, 1999). In what follows, we first introduce this framework in the simpler case where no dependent clause appears in the sentence.

In the 'implicit' comparison (17), what is compared to must be recovered from the use context; this is indicated by the free variable δ, standing for some degree. The interpretation of this sentence is read, 'there is an event e in which the steak sizzles, and e is appetizing to a degree greater than δ'.

(17) The steak sizzled more appetizingly. $\rightsquigarrow$
$\exists e[Th(e)(s)$ & **sizzle**(e) & **appetizing**$(e) > \delta]$

When the dependent clause is present, the combination of ellipsis resolution and semantic composition delivers a degree that takes the place of δ in a representation like that in (17). (18) is read as, 'the maximal degree d to which there is an event e of the hamburger sizzling, and e is appetizing to at least degree d'. Semantically, the maximal degree ($max\ d$) is introduced by a null operator that we will call *how* (Kennedy, 2002) throughout this paper.

(18) ...than the hamburger did. $\overset{resolve\ ellipsis}{\rightsquigarrow}$

(19) ...the hamburger did$_{\text{sizzle } \textit{how}\text{-appetizingly}}$ $\rightsquigarrow$
$max\ d.\exists e[Th(e)(h)$ & **sizzle**(e) & **appetiz**$(e)\ \geq d]$

Putting the pieces together, (13) in fact has the richer and more accurate meaning representation in (20).

(20) $\exists e_1[Th(e_1)(s)$ & **sizz**(e_1) & **appetiz**$(e_1) > max\ d.]$
$\exists e_2[Th(e_2)(h)$ & **sizz**(e_2) & **appetiz**$(e_2) \geq d]$

In comparatives with *more*/*-er*, and equatives with *as*, how the 'scale' is introduced in the dependent clause differs according to the major part of speech of the comparison structure. For adjectival and adverbial comparisons (*taller*, *as quickly*), the scale is provided by those categories (height, appetizingness) and the null operator is simply *how*. For nominal and verbal comparisons (*more rice*, *sizzle as much*), *much* introduces a variable scale (μ), and the null operator is called *how-much*.

Related Work

In addition to the major characteristics pointed out in the paper, our framework improves on the following issues as compared with Bakhshandeh and Allen (Bakhshandeh and Allen, 2015):

– While we also model comparison structures as predicate-argument pairs, we do not use additional semantic role links. We retain all semantic information on predicate and argument types, which results in better semantic generalization across all predicates (Section 3).

– We categorize arguments into semantic frames associated with each predicate type. This enables addressing complex cases such as 'copulas' (Section 3.1.2) which play a crucial role in asserting properties about entities. Furthermore, we introduce a more comprehensive set of argument types which more accurately capture the syntactic and semantic properties of various predicate types.

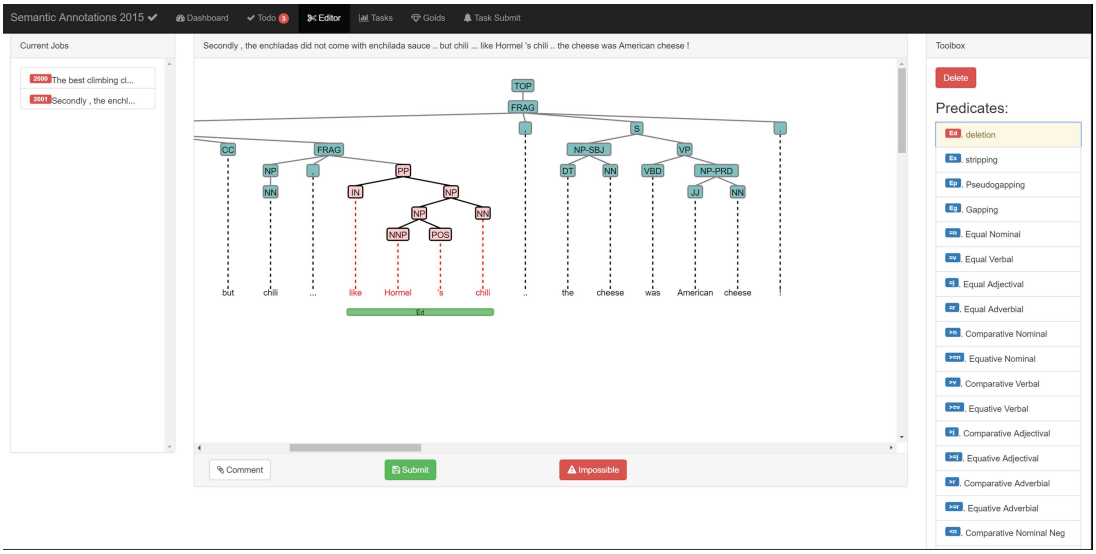


Figure 4: A screen-shot of our two-stage tree-based annotation tool.

Integer Linear Programming Constraints

Overall, our ILP constraints (which encode restrictions on the arguments of predicates) are either applied in general (to any predicate type) or are tailored to encode the semantic frame of a specific predicate. Following are our generic constraints:

1. The maximum number of arguments per node is 3.
2. The maximum number of arguments in the entire syntactic tree is 10.

We incorporate the following ILP constraints for encoding knowledge regarding Ellipsis predicates:

1. The constituency span of comparison predicate's $Figure$ and $Ground$ should overlap with the $Reference$ argument of ellipsis predicate, if any.
2. The constituency node of $Exclude$ argument should be a child of the $Reference$.
3. One node can only have more than one comparison argument type if those types are $Figure$ and $Ground$.

The constraints for encoding the semantic frame of the other comparison predicate types follows straightforwardly from the semantic frames presented in the paper.

Data Collection Methodology

One approach for extracting sentences containing comparisons is to mine the text for some (automatically or manually created) patterns, then train a classifier for labeling comparison and non-comparison sentences (Jindal and Liu, 2006b). However, the variety of comparison structures is so vast that being limited to some specific patterns or syntactic structures will not result in good coverage of comparisons. Instead, we use the following filter ($CompFilter$) with a set of basic comparison structure linguistic markers for extracting potential comparison instances:

– Any sentence containing a word with POS tag equal to *JJR*, *RBR*, *JJS*, or *RBS*.

– Any sentence containing a comparison morpheme such as *more*, *most*, *less*, *enough*, *too*.

This filter is guaranteed not to have any false negatives since it is exhaustive enough to capture any possible comparison sentence. We applied this filter to the English Web Corpus and the Movie Reviews dataset and extracted a pool of 2,800 sentences for final annotation in the next step. It is important to note that this filter will capture some cases which look like comparison instances at the surface level, but which are not so semantically (e.g., (21)-(22), extracted from the Google Web Treebank). Such negative examples help the quality of the final prediction models.

(21) Very nice ambiance and friendly staff *too*.

(22) We had sesame chicken and kung pao chicken *as* well as cheese puffs.

Baseline Model

We implemented a rule-based baseline for predicate-argument structure prediction. This model mainly uses POS and lexical wording rules for predicate prediction. For example, we have the

following rule for predicate prediction: Any JJS POS tag can be tagged as a superlative predicate.

For argument prediction, we mainly implement our knowledge-driven ILP constraints as rules. Furthermore, this baseline uses rules such as the following: in any *than-clause*, the first NP should be tagged as Ground argument. Also, the subject (if any) should be tagged as Figure argument, and the closest adjective to the comparison morpheme is the Scale indicator.

Two-stage Tree-based Annotation Tool

We are releasing our interactive two-stage tree-based annotation tool with this paper. In this tool each annotator can be assigned with a set of tree-based annotation assignments, where pairing annotators to do the same task for inter-annotator analysis is also feasible. This annotation tool sets up the data collection as a two-stage expert annotation process: (1) for each sentence, one expert annotates and submits the annotation, (2) another expert reviews the submission and either returns the submission with feedback or marks it as a gold. Figure 4 shows a screen-shot of this tool.

References

James F. Allen, Mary Swift, and Will de Beaumont. 2008. Deep semantic analysis of text. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, STEP '08, pages 343–354, Stroudsburg, PA, USA. Association for Computational Linguistics.

Omid Bakhshandeh and James Allen. 2015. Semantic framework for comparison structures in natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 993–1002, Lisbon, Portugal, September. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Association for Computational Linguistics (ACL)*.

Johan Bos and Jennifer Spenader. 2011. An annotated corpus for the analysis of vp ellipsis. *Language Resources and Evaluation*, 45(4):463–494.

Johan Bos. 2008. Wide-coverage semantic analysis with boxer. In Johan Bos and Rodolfo Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings*, Research in Computational Semantics, pages 277–286. College Publications.

Joan Bresnan. 1973. Syntax of the comparative clause construction in English. *Linguistic Inquiry*, 4(3):275–343.

Max Cresswell. 1976. The semantics of degree. *Barbara Hall Partee (ed.)*, pages 261–292.

Dipanjan Das, Desai Chen, AndrÃl' F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40:1:9–56.

Donald Davidson. 1967. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, pages 81–95. Pittsburgh University Press, Pittsburgh.

Dan Flickinger. 2011. Accuracy vs. robustness in grammar engineering. In Emily M. Bender and Jennifer E. Arnold, editors, *Language from a Cognitive Perspective: Grammar, Usage and Processing*, number 201, pages 31–50. CSLI Publications, Stanford.

Daniel Hardt. 1997. An empirical approach to vp ellipsis. *Comput. Linguist.*, 23(4):525–541, December.

Nitin Jindal and Bing Liu. 2006a. Identifying comparative sentences in text documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 244–251, New York, NY, USA. ACM.

Nitin Jindal and Bing Liu. 2006b. Mining comparative sentences and relations. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pages 1331–1336. AAAI Press.

Chris Kennedy. 1999. *Projecting the Adjective: The Syntax and Semantics of Gradability and Comparison*. Garland, New York.

Chris Kennedy. 2002. Comparative deletion and optimality in syntax. *Natural Language and Linguistic Theory*, 20(3):553–621.

Christopher Kennedy. 2003. Ellipsis and syntactic representation. In Kerstin Schwabe and Susanne Winkler, editors, *The Interfaces: Deriving and Interpreting Omitted Structures*, number 61 in Linguistics Aktuell, pages 29–54. John Benjamins.

Wiltrud Kessler and Jonas Kuhn. 2014. Detecting comparative sentiment expressions – a case study in annotation design decisions. In *Proceedings of KONVENS*, Hildesheim, Germany.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Jason Merchant. 2013. Voice and ellipsis. *Linguistic Inquiry*, 44(1):77–108.

Leif Arda Nielsen. 2004. Verb phrase ellipsis detection using automatically parsed text. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124.

Terence Parsons. 1990. Events in the semantics of English: A study in subatomic semantics. In *Current Studies in Linguistics Series no. 19*, page 334. MIT Press, Cambridge, Massachusetts.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 433–440, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michael Schiehlen. 2002. Ellipsis resolution with underspecified scope. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 72–79, Stroudsburg, PA, USA. Association for Computational Linguistics.

Arnim Von Stechow. 1984. Comparing semantic theories of comparison. *Journal of Semantics*, 3(1):1–77.

Alexis Wellwood. 2015. On the semantics of comparison across categories. *Linguistics and Philosophy*, 38(1):67–101.

Kaiquan Xu, Stephen Shaoyi Liao, Jiexun Li, and Yuxia Song. 2011. Mining comparative opinions from customer reviews for competitive intelligence. *Decision Support Systems*, 50(4):743–754.

Masaya Yoshida, Chizuru Nakao, and IvÃąn Ortega-Santos. 2016. *Ellipsis*. Routledge Handbook of Syntax, London, UK.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, pages 1050–1055. AAAI Press.