Language and Cognition **9** (2017), 254–292. doi:10.1017/langcog.2016.7

Computational learning of construction grammars*

JONATHAN DUNN
Illinois Institute of Technology, Department of Computer Science

(*Received* 12 *January* 2016 – *Revised* 04 *February* 2016 – *Accepted* 10 *February* 2016 – *First published online* 28 *March* 2016)

ABSTRACT

This paper presents an algorithm for learning the construction grammar of a language from a large corpus. This grammar induction algorithm has two goals: first, to show that construction grammars are learnable without highly specified innate structure; second, to develop a model of which units do or do not constitute constructions in a given dataset. The basic task of construction grammar induction is to identify the minimum set of constructions that represents the language in question with maximum descriptive adequacy. These constructions must (1) generalize across an unspecified number of units while (2) containing mixed levels of representation internally (e.g., both item-specific and schematized representations), and (3) allowing for unfilled and partially filled slots. Additionally, these constructions may (4) contain recursive structure within a given slot that needs to be reduced in order to produce a sufficiently schematic representation. In other words, these constructions are multi-length, multi-level, possibly discontinuous co-occurrences which generalize across internal recursive structures. These co-occurrences are modeled using frequency and the ΔP measure of association, expanded in novel ways to cover multi-unit sequences. This work provides important new evidence for the learnability of construction grammars as well as a tool for the automated corpus analysis of constructions.

KEYWORDS: construction grammar, grammar induction, multi-unit association measures, poverty of the stimulus.

1. Learning construction grammars

The Cognitive Linguistics paradigm holds that language is not strictly separated from other cognitive faculties (e.g., Langacker, 1987; Hilpert, 2008) and, to some

[*] The author would like to thank Shlomo Argamon and Joshua Trampier for their support and engagement throughout this project. This work was funded in part by the Oak Ridge Institute for Science and Education. Address for correspondence: 3300 South Federal Street, Chicago, IL 60616; web: www.jdunn.name; e-mail: jonathan.edwin.dunn@gmail.com

degree following from this, that languages are learnable without highly specified innate structure (e.g., Hopper, 1987). That is, languages are learnable from the statistical properties of observed linguistic expressions without positing innate structures present in the learner (e.g., Goldberg, Casenhiser, & Sethuraman, 2004; Bybee, 2006; Goldberg, 2009). A 'Grammar' within Cognitive Linguistics, then, is a data-driven and ultimately domain-independent model able to learn grammatical generalizations from linguistic input. More precisely, any innate constraints on the Grammar in this paradigm are not specific to language but rather are general cognitive constraints (e.g., limits on working memory, ability to recognize and categorize differences, etc.) that, when applied to language learning, result in cross-linguistic patterns. One argument advanced for innate structure is that language learners are exposed to different instances of observed language but reach relatively similar grammatical representations. The question, then, is whether this stability results from learners sharing a partially defined initial state (e.g., innate structure) or from learners sharing a single domain-independent ability to generalize from observations.

A lower-case grammar is the representation of a specific language while an upper-case Grammar is the ability to learn such a grammar from linguistic input alone with minimal innate structure. Thus, language-specific construction grammars (e.g., analyses in Fillmore, 1988, and Kay & Fillmore, 1999) can be seen as part of a more general Construction Grammar (e.g., Goldberg, 2006; Langacker, 2008). This differs from Chomsky's various divisions of competence/performance and universal/specific grammar (1965, 1975), however, in that the Grammar does not consist of predefined structures/rules/constraints but rather of mechanisms for deriving or learning such structures/rules/constraints from observed language data. This data-driven view can be visualized as in Figure 1, where the Grammar is a link between language observations and generalized language representations (grammars).

This illustration of the data-driven view of Grammar should not be mistaken for an innate Language Acquisition Device (e.g., Briscoe, 2000). The view here is that the Grammar consists largely or entirely of domain-independent principles for deriving generalizations from a series of observations, and that the form of produced grammars is a result of (i) the observed language data itself and (ii) the domain-independent principles for forming generalizations. In other words, from this perspective Grammar "is not an overarching set of abstract principles, but more a question of a spreading of systematicity from individual words, phrases, and small sets" (Hopper, 1987, p. 142). This implies, for example, that a speaker's grammar is not fixed but rather continues to be modified as more language use is observed. The essential difference between these views is whether systematicity in language is seen as a top-down phenomenon (defined by innate structure) or a bottom-up phenomenon (defined by spreading systematicity from observed language use).

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

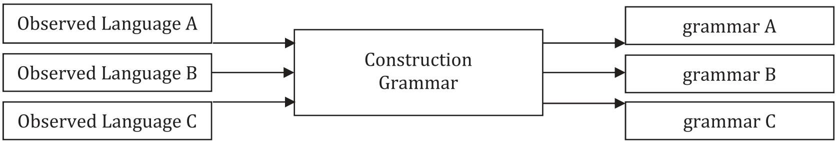

Fig. 1. Grammar and grammars.

The debate over an innate or a data-driven language faculty comes down in part to a simple empirical question: Is it possible to learn the grammar of a language without innate structure? In other words, is Grammar a set of structures or a set of mechanisms for learning such structures? This question has been approached with a variety of evidence; the point of this paper is to provide computational corpus-based evidence by simulating the language-learning process with computational models (e.g., Goldsmith, 2001, 2006; Solan, Horn, Ruppin, & Edelman, 2005; as opposed to the approach taken in Briscoe, 2000). If a grammar-induction algorithm is capable of learning the grammar of a language without innate structure and using purely statistical properties of observed language data, then it follows that such grammar learning is possible in principle given only linguistic input. This is the case even though the model is provided written language while human learners are provided spoken language, and even though human and computational learners do not employ the same mechanisms. In other words, the question is whether the regularities of language can be adequately generalized into a productive model of grammar given only observed 'surface' linguistic expressions.

Katzir (2014) observes that such computational simulations can be a counter-argument to the poverty-of-the-stimulus line of reasoning for Universal Grammar. However, this does not address either the richness-of-the-stimulus or typological lines of reasoning for Universal Grammar. Thus, this is one piece among many for the view of language as a learned phenomenon. It is, further, only one piece of converging evidence against the poverty-of-the-stimulus line of reasoning. For example, there are two main weaknesses to this source of evidence: (i) that the algorithm requires access to much more data than do human learners, and (ii) that that data is presented all at once rather than being observed sequentially across many occasions. We can perhaps divide the poverty-of-the-stimulus argument into two parts: first, that language cannot be learned without innate structure as a matter of quality of observations, in part because only positive examples can be observed; second, that language cannot be learned without innate structure as a matter of quantity of observations, in that language learners have access to different amounts of linguistic input but reach similar grammatical representations. This source of evidence, then, deals only with poverty-of-the-stimulus in terms of quality of observed language and not in terms of quantity.

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

This work can also be seen as a response to criticisms (e.g., Bod, 2006) that construction grammar makes imprecise and thus untestable predictions. In other words, it provides a reproducible model of which possible constructions qualify as actual constructions in reference to a given dataset, a question that is not adequately addressed in the literature. Section 1 of this paper examines the nature of a construction grammar, the definition of a construction, and the properties of constructions which the model must capture. Section 2 describes the grammar induction algorithm in detail. Section 3 presents several introspective and quantitative evaluations of the output grammar for subsets of the ukWac corpus of web-crawled English (Baroni, Bernardini, Ferraresi, & Zanchetta, 2009).

1.1. GRAMMAR AS MEANINGFUL SYMBOLIC UNITS

The basic idea of construction grammar is that grammar is more than simply a formal system consisting of stable but arbitrary rules for defining well-formed sequences. Grammar, instead, consists of meaningful and symbolic form–meaning mappings, called constructions (Goldberg, 2006). In this sense, a grammar consists of meaningful constructions in the same way that a lexicon consists of meaningful words (Langacker, 2008). The task of learning the grammar of a language, in this paradigm, is the task of learning the vocabulary of meaningful symbolic units which makes up that grammar. This allows us to bring together two important premises: first, that grammar consists of meaningful symbolic units (e.g., Langacker's Cognitive Grammar); second, that co-occurrence and distribution are indicators of meaning (e.g., Firth, 1957).

Taken together, these premises suggest that constructions, like words, can be studied and defined as a set of co-occurring elements in a corpus. In this case, however, the elements are in fact abstract and productive schemas representing a large number of linguistic forms. If grammars consist of symbolic form–meaning mappings, and if the distribution of elements in a corpus reveals their meaning, then the problem of grammar induction can be viewed as the problem of distinguishing those potential constructions which significantly co-occur from those potential constructions which do not significantly co-occur. It should be noted that the constructions discussed here are not simple idioms or phrases (although such may be constructions) but rather range from fully schematic and productive to fully item-specific representations.

Given that constructions are productive and co-occurring schemas, such co-occurrences can be disguised in observed language data by two sorts of phenomena: (1) by recursive structure within a particular element of the construction; or (2) by unfilled or partially filled elements in the construction. This means that the language represented by these constructions can appear

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

to be discontinuous. The problem is that this greatly increases the hypothesis space and raises the question of learnability: With such a large hypothesis space, unconstrained by innate structure, is it feasible for the learner to distinguish valid constructions from the much larger number of potential constructions? Lidz and Williams (2009), for example, argue that the great number of possible grammatical representations, taken together with similar learned output grammars across language learners, requires the constraining power of innate structures/rules/constraints. This objection is countered in the evaluation section of this paper by comparing the agreement of output grammars learned from different subsets of the corpus. In this case, the subsets represent multiple learners with the same Grammar learning the language from different inputs.

1.2. PRECISE DEFINITIONS FOR WHAT CONSTITUTES A CONSTRUCTION

What is a construction? Or, asked another way, which units count as grammatical entities (i.e., constructions) for a given speaker and a given language? The discussion above contrasts potential constructions and actual constructions, framing the language-learning task as one of distinguishing between these two categories. A Construction Grammar in the sense discussed here provides a mathematical definition of co-occurrence such that the theory can distinguish between potential and actual constructions and thus produce a set of constructions (i.e., a grammar) representing a given language. This sort of grammar is updateable in the sense that the units which qualify as constructions change over time as new language use is observed. The model is based on form (e.g., multi-length and multi-level non-continuous sequences with possible internal recursive structure) and distribution (e.g., frequency and multi-unit association measures). The implicit hypothesis, then, is that constructions can be identified using these measures on surface linguistic expressions.

A counter-argument to this program of precisely defining constructions is that it is based on the classical theory of categorization's strict category boundaries rather than on the fuzzy and gradient membership posited by proto-type categorization theory. This is a false dichotomy, however, because the model ranks constructions using scalar measures. The classical, strict categorization approach can be simulated by setting a strict boundary threshold. The proto-type, fuzzy categorization, approach can be achieved by retaining the order of constructions posited by the model. In short, the container metaphor for language (e.g., that a grammar and a lexicon contain certain elements and not others) is a conventional way of discussing linguistic theory, even when we are aware that parts of this metaphor are not accurate (e.g., Langacker, 2006). In other words, the idea of an optimum grammar

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

to describe a language is a metaphoric idea, subject in practice to variations within speakers (e.g., across genres) and between speakers (e.g., across speech communities). Although not explored further here, such variations in learned construction grammars occur at two levels: types of constructions (presence or absence of a given construction) and usage of constructions (relative frequency of a given construction).

The grammatical generalizations learned by the algorithm are abstracted away from individual speakers by definition, in that they are learned from a corpus of data produced by many speakers. Thus, the argument presented here participates in the abstraction by which language-use is generalized away from individuals and discussed as a single entity such as ‘English’ or ‘German’. This abstraction means that the elements of a grammar are not necessarily a psycholinguistic reality for any single speaker, a limitation that also applies to the work presented here.

1.3. PROPERTIES OF CONSTRUCTIONS TO BE MODELED

Constructions are form–meaning mappings that differ in their size, internal complexity, and level of schematicity. This paper is concerned only with constructions above the level of individual words. The constructions that need to be identified are idioms like the partially filled idiom in (1), argument constructions like the ditransitive in (2), and sentence-level constructions like the covariational conditional in (3) (c.f. Goldberg, 2006).

(1) jog [someone’s] memory
(2) NP + <TRANSFER> + NP + NP
(3) the [X’er], the [Y’er]

These examples represent three of the essential properties of constructions that need to be captured: (i) varying length, (ii) varying levels of representation in each slot, and (iii) filled, partially filled, or unfilled slots. A fourth essential property of constructions (iv) is the ability to contain recursive material within a given slot (e.g., a nominal construction nested within a verbal construction) as well as constituents with varied internal structure.

The first challenge is that constructions vary in length and that word-based measures of length do not account for constituent-internal structure. For example, the idiom in (1) contains three units, while the ditransitive in (2) contains four units. Further, and creating a greater difficulty, constructions can have recursively filled slots. For example, (4a) through (4c) contain instances of the same ditransitive construction but contain different numbers of lexical units, ranging from five to eight. The algorithm must be able to generalize over these different lengths and recursively filled slots to identify the underlying construction: NP + <TRANSFER> + NP + NP. In other words,

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

co-location can occur at the word-level but also at the phrase-level, so that in (4c), for example, *Bill's uncle* and *two Canadian dollars* can be seen as being separated by six units (at the word-level) or by two units (at the phrase-level). The algorithm must be sufficiently flexible to allow item-specific representations (e.g., (4e)) to be identified alongside fully schematized representations as in (2). In other words, the problem is how to measure multi-level co-occurrence.

(4) a. Bill gave Wendy two dollars.
b. Bill gave Wendy's sister two dollars.
c. Bill's uncle gave Wendy's older half-sister from Paris two Canadian dollars.
d. Bill's uncle gave Wendy a hand.
e. gave X a hand

The second challenge is that constructions vary in the level of representation used and may contain mixed levels of representation. For example, the ditransitive construction in (2) must be represented using parts-of-speech and semantic categories. The idiom in (1), on the other hand, has to be represented at multiple levels: the fixed part of the idiom requires simple lexical representation but the unfilled slot has semantic restrictions (e.g., an animate object). This multi-level requirement makes the task more difficult than collocation identification and, more importantly, again multiplies the space within which the learner must search for potential constructions.

The grammar induction algorithm operates on three levels: first, on lemmatized word-forms representing the lexical level of language; second, on part-of-speech forms representing lexical units grouped according to their syntactic distribution; third, on semantic or conceptual forms representing lexical units grouped according to their meaning. In addition, the algorithm allows for the reduction of internal structure within prepositional phrases, noun phrases, multi-word named entities, and adjunct units in order to measure distance at both the fully schematized and the item-specific levels for purposes of measuring co-occurrence. These phrasal representations are similar to Fillmore's (1988) 'maximal' categories, whereas the lemma and part-of-speech representations are similar to 'minimal' categories.

The third challenge is that constructions contain filled, partially filled, and unfilled slots. In other words, a particular slot of the construction can be filled by a lexical item, can be constrained to a unit of a particular semantic category, or can be left entirely unfilled. This means that a construction can be non-continuous in the surface linguistic expression. For example, the idiom in (5) has an unspecified slot which, however, must be filled by a human or some entity which takes on the properties of a human via metonymy or personification. The idiom in (6a), however, can be filled by any material

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

whatsoever, as shown by the examples in (6b–d). The algorithm deals with this requirement by using multiple levels of representation: partially filled slots can be defined by their semantic requirements (e.g., any animate object), and unfilled slots can be defined by their syntactic requirements (e.g., any noun phrase). This again multiplies the search space for potential constructions.

(5) send [SOMEONE] to the cleaners
(6) a. They didn't pay [NP] any heed.
b. They didn't pay [me] any heed.
c. They didn't pay [the warning signs] any heed.
d. They didn't pay [the smoke on the horizon] any heed.

The fourth challenge is that constructions can have recursively filled internal structure. This takes two forms: (i) a syntactically defined slot can be filled with a wide range of complex constituents of the same type (e.g., NPs take many different forms), and (ii) constructions can be nested within other constructions. As an example of the first case, if we take the ditransitive construction in (2) above, repeated in (7a), any of the components can contain constituents with varied internal structure, so that (7b) through (7d) are all instantiations of the same construction. As an example of the second case, (7e) contains the same ditransitive construction nested within a different instance of the construction, so that *ball* is part of the main ditransitive as well as the relative clause version of the ditransitive. The first sort of recursion, of interchangeable constituents in a single more general slot, although a challenge to model, is a relatively simple phenomenon for construction grammar in general. The second sort, however, is more difficult on both levels.

(7) a. NP + <TRANSFER> + NP + NP
b. He gave her the ball.
c. The short man quickly gave her the blue ball.
d. The two short men quickly refused to give her any of the balls.
e. He gave her the ball Bob had just given him two days before.

The constructions output by the algorithm have a linear form such as in (8a–d). In this formula, units of a given level of representation occur in the specified order. Four levels of representation are used in the final output: first, specific word-forms and lemmas, as in (8a) with "be"; second, part-of-speech tags for individual units, as also in (8a) with the units in brackets; third, semantic or conceptual categories which constrain the fillers of the slot in question, as in (8c) in small caps; fourth, syntactic phrases with reduced internal structure, such as NP and PP in (8d).

(8) a. [Wh-Determiner] + [Modal] + "be" + [Past-Participle]
b. "to" + [Verb] + [Determiner] + [Noun]

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

c. [Noun] + [Preposition] + [Determiner] + <PLANNING>
d. "be" + [Past-Participle] + PP+ NP

The use of multiple levels of generality shows the influence of corpus linguistics on the algorithm in addition to Cognitive Grammar: the goal is to find the inventory of symbolic grammatical units attested in the corpus, even if those units are not abstract or schematic but rather fully item-specific. This is an important part of grammar induction because observed patterns in usage show that speakers have clear preferences both for schematic structures and for specific instances of such structures.

Finally, an essential property of constructions more generally is that they are form–meaning mappings rather than purely syntactically defined sequences. This is modeled here both directly and indirectly. Directly, it is captured using semantic or conceptual representations of words; in effect, this means that the filler of a slot can be defined in terms of a specific meaning, rather than in terms of a specific lexical or syntactic item. Indirectly, this is captured using overlapping constructions with different levels of schematicity. More item-specific constructions represent different instances of a more general or schematized construction and have different meanings from generic instances of that construction (e.g., *give me two pieces of cheese* vs. *give me a hand*).

2. The construction induction algorithm

This section looks at the construction induction algorithm[1] in detail, starting in Section 2.1 with a discussion of the underlying problem and how it is distributed across the algorithm. Section 2.2 looks at the different levels of representation used in the algorithm. The core functions of the algorithm are then examined: the generation of potential constructions (2.3), formulating association measures to evaluate candidates (2.4), and then using association measures to select the best candidates. The algorithm is then situated relative to other computational work on constructions, relative to collostructional analysis, and relative to other work on grammar induction (2.5).

2.1. ASPECTS OF THE PROBLEM

The goal of the construction grammar induction algorithm is to search through the many linguistic expressions present in a large corpus in order to find the relatively small number of underlying generalizable grammatical units which produce or represent those linguistic expressions. In other words, the problem

[1] Code and related data for the Construction Induction algorithm is available at www.jdunn.name.

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

is to cut through the noise in the textual data and return only those units which can be considered part of the grammar represented in the corpus. The linguistic expressions in the corpus have a very large number of possible representations (i.e., potential constructions); the problem is to find the optimum set of representations.

The construction grammar induction algorithm identifies multi-length, multi-level, non-continuous co-occurrences while abstracting over internal recursive structure. In other words, the algorithm builds frequency and association measures of co-occurrence but does so at multiple levels of analysis. This task is divided across three stages in the algorithm: first, the candidate generation stage deals with recursive structures and non-continuous representations. Second, the construction identification stage forms templates for construction types and identifies the presence of these construction templates in linguistic expressions in order to extract and inventory potential constructions. Third, the candidate evaluation stage searches through the very large number of potential grammatical representations (i.e., candidate constructions) to determine the set which best represents the linguistic expressions in the input corpus using frequency and multi-unit association measures. The pseudo-code for the algorithm is shown in Table 1; this pseudo-code can be considered a diagram of the essential workings of the algorithm and also a guide to a specific Python implementation.

2.2. LEVELS OF REPRESENTATION

Level of representation refers to the type of linguistic analysis used to label a particular element in the construction: part-of-speech (e.g., noun), phrase type (e.g., prepositional phrase), semantic-category (e.g., animate), and lemma (e.g., "candle"). The idea behind varying levels of representation within a construction is (1) that language is composed of layered and interacting levels of structure and (2) that grammatical units can be fossilized at each level. In other words, some constructions may be completely schematic and others may be completely item-specific. The algorithm, therefore, must operate on multiple levels of representation because we cannot know a priori for a given linguistic expression the specificity or type of representation present in the construction that produced it.

The algorithm has a few dependencies. First, it relies on part-of-speech tagging (in this case, TreeTagger: Schmid, 1994), which labels lexical units according to their syntactic distribution and function. Second, it relies on semantic or conceptual tagging (in this case, the UCREL Semantic Analysis System: Piao, Bianchi, Dayrell, D'Egidio, & Rayson, 2015), which labels lexical units according to their ontological meaning. Third, it relies on a dependency parser (in this case, MaltParser: Nivre et al., 2007), which aids

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

TABLE 1. *The construction-grammar induction algorithm*

```
1    Create unit inventories for each level of representation
       a. Create list of all unit values at each level of representation
       b. Discard unit values below frequency threshold
       c. Assign each unit value a numeric index
2    Ingest input files
       a. Divide into units divided by sentence boundaries and/or punctuation (by parameter)
          i. Represent each unit as vector of unit value indexes
          ii. Represent each clause/sentence as a collection of unit vectors
3    Search for recursive structures and non-continuous units
       a. For each clause:
          i. Look for adjunct units (e.g., adverbs)
          ii. Look for PPs (e.g., "into the house")
          iii. Look for NPs (e.g., "the house")
          iv. Look for Multi-Word Named Entities (e.g., "Norman Rockwell")
       b. For each reduction in each clause:
          i. Create alternate clause with unit either reduced (e.g., to "NP") or removed
          ii. Create alternate clauses with all combinations of reductions applied
4    Create construction templates
       a. For all lengths from 2 through N (Max construction length):
          i. All possible combinations of levels of representation
5    Extract candidate constructions using templates and units of text
       a. For each template:
          i. Search through original and alternate linguistic expressions
          ii. Extract and count all matches
          iii. Disregard any matches containing discarded labels
          iv. Remove all candidates below the frequency threshold
6    Evaluate candidates:
       a. Frequency
       b. Summed ΔP, Left-to-Right
       c. Summed ΔP, Right-to-Left
       d. Mean ΔP, Left-to-Right
       e. Mean ΔP, Right-to-Left
       f. Beginning-Reduced ΔP, Left-to-Right
       g. Beginning-Reduced ΔP, Right-to-Left
       h. End-Reduced ΔP, Left-to-Right
       i. End-Reduced ΔP, Right-to-Left
       j. Beginning-Divided ΔP, Left-to-Right
       k. Beginning-Divided ΔP, Right-to-Left
       l. End-Divided ΔP, Left-to-Right
       m. End-Divided ΔP, Right-to-Left
       n. Direction Scalar ΔP
       o. Direction Categorical ΔP
7    Prune candidates:
       i. By Association Strength
       ii. Horizontally (prefer longest candidates)
       iii. Vertically (remove alternate representations)
```

in the reduction of prepositional phrases and noun phrases. There is no theoretical reason why these functions could not be incorporated into a single framework, only the practical consideration of avoiding the duplication of existing work. These dependencies do not invalidate the argument against

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

innate structure because each could itself be performed in an unsupervised and data-driven fashion.[2]

2.3. GENERATING POTENTIAL CONSTRUCTIONS

The candidate generation step carries the weight of deriving possible generalizations from each linguistic expression. There are two separate stages here: first, producing alternate representations of a linguistic expression to reduce recursive units; second, extracting construction templates of varying length and level of representation from those alternate representations of the linguistic expressions (i.e., steps 3–5 in the pseudo-code).

For example, the sentences in (9a–c) all depend on the ditransitive construction, with increasing substructures within the slots of the construction that create noise for the language-learning algorithm. In other words, finding the construction "NP + <TRANSFER> + NP + NP" from the sentence in (9c) requires looking at each constituent as a whole, as shown with brackets in (9d). The algorithm approaches this problem by generating alternate forms for each linguistic expression and then including these alternate forms in the search for co-occurrences.

(9) a. "The coffee gave her a headache."
b. "The dark unfiltered coffee soon gave her a splitting headache."
c. "The dark unfiltered coffee from South America soon gave her a splitting headache and a feeling of nausea."
d. "[The dark unfiltered coffee from South America] [soon gave] [her] [a splitting headache and a feeling of nausea]."

Given an expanded set of linguistic expressions, the algorithm handles varying length and varying levels of representation by creating templates for all possible combinations of representations within the defined length parameter. Each template, therefore, represents the most abstract properties of a construction: How many units and what representations does it contain? The algorithm then extracts all potential constructions, which are simply instantiations of each template in a linguistic expression.

2.4. EVALUATING POTENTIAL CONSTRUCTIONS

The evaluation of potential constructions involves mathematically modeling the properties which separate constructions and non-constructions, either

[2] More recent versions of the algorithm incorporate a distributional method of creating semantic dictionaries as well as the unsupervised learning of phrase structure rules which supports the further reduction of complex constituents, thus removing two of the three dependencies.

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

with a sharp delineation of the two categories or with a scalar ordering by degrees of entrenchment. In this case, the model is observational in that it operates on a corpus of attested linguistic expressions. Thus, the question is what quantitative distributional measures are required to develop a model of constructions. Two standard measures are used: frequency and association strength. The implementation of these standard measures, however, must allow for the evaluation of multi-unit candidates, which requires developing multi-unit association measures.

The first measure is frequency, a simple representation of how often something appears in the dataset. This measure is relative frequency, in that all candidates are evaluated on the same dataset. In addition to providing a constraint on the overall search space, frequency remains an important measure of a candidate's status as a construction, in order to prefer some possible representations over others. The frequency threshold is enforced by creating an index of unit frequencies on the entire corpus or on a significant subset of the corpus (i.e., a million word subset) and ignoring those units which do not pass this indexing threshold. While this reduces the search space for the algorithm, it is not psychologically plausible in the sense that human learners do not have this sort of large existing dataset to query in advance of learning. As noted in more detail below, one critical assumption behind this approach is that human learners have the ability to store and update the frequencies of units and sequences of units largely without limit. The present algorithm, because it has access to the entire corpus all at once, can use frequency indexing as a means of reducing the hypothesis space in a way that human learners cannot.

Association strength is measured using the bi-directional ΔP (Gries, 2013; cf. Gries, 2008, 2012), calculated both left-to-right and right-to-left, as shown in Table 2. To be more precise, the ΔP is not bi-directional but rather consists of two direction-dependent measures; taken together, these two direction-dependent measures allow us to model linguistic associations in all possible directions. Both spoken and written language are one-dimensional in the sense that Unit A can either come before or come after Unit B. The construction induction algorithm is based on multi-directional (left-to-right or right-to-left), multi-dimensional (across varying levels of representation), multi-length (across two or more units) association strength, measured with and without complex constituent-internal structure (i.e., distance is measured at different levels of abstraction). The idea is that sequences which are constructions (e.g., are cognitively entrenched to some degree) are more internally associated than sequences which are not constructions (e.g., those which are chance co-occurrences of units). The purpose of the association measures (and the frequency counts on which such measures are ultimately based) is to learn an inventory of constructions from the very large hypothesis space of all observed sequences.

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

TABLE 2. *Calculating ΔP*

1	Let X be a unit of any representation
2	Let Y be any other unit of any representation
3	Let X_a indicate that unit X is absent
4	Let X_p indicate that unit X is present
5	$\Delta P(X \mid Y) = p(X_p \mid Y_p) - p(X_p \mid Y_a)$
6	$\Delta P(Y \mid X) = p(Y_p \mid X_p) - p(Y_p \mid X_a)$

Like most linguistic association strength measures, ΔP is usually employed to measure the relationship between two individual words. Given the variable length required by constructions, this is converted into a multi-word measure in four different ways. Each calculation is given for a sequence of elements listed in (10) for the sake of example. Association strength is an important addition to frequency because it allows the model to capture the constraint of degree of openness (Goldberg, 2006). The basic problem is that very frequent units occur often in competing potential constructions and association measures prevent the over-identification of false positive constructions containing frequent units.

(10) A B C D E F

First, the simplest multi-word measure is a sum of the total directional association within a candidate, implemented with a minimum pairwise threshold. In other words, so long as each pairwise ΔP is above the threshold, this measure simply sums the total association strength. While this first measure tends to favor longer candidates, it is left as-is in order to counteract the frequency thresholds which tend to favor shorter candidates. An alternate version, the mean ΔP, is normalized by the length of the candidate in number of units to produce the mean pairwise association score across the entire sequence. Both measures are shown in Table 3.

This multi-unit measure is similar to Daudaravičius and Marcinkevičienė's (2004) work on detecting the borders of collocations, except that it allows both a minimum threshold and a final score (e.g., the summed association strength). In other words, the gravity count measure is a different formulation for association strength and a collocation is defined as a sequence of pairs whose association falls above a given threshold. The summed ΔP is similar, except that it also outputs a sum of pairwise associations for those sequences which do exceed the threshold. This similarity is disguised by a difference in implementation. For example, Jelinek (1990) also uses an iterative approach that tests increasingly longer sequences for sufficient association strength; in the current implementation, each candidate is considered independently, although any longer sequence which passes the frequency threshold is by definition made up of smaller sequences which have themselves passed that threshold.

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

TABLE 3. *Calculating the Summed ΔP*

1	Calculate each ordered pairwise ΔP:
2	A|B, B|C, C|D, D|E, E|F
3	F_p = Pairwise Frequency Threshold
4	If any ordered pairwise ΔP < F_p, discard candidate construction
5	Summed ΔP = Σ(ΔP(A|B))(ΔP(B|C))…
6	Mean ΔP = (Σ(ΔP(A|B))(ΔP(B|C))…) / N_{units}

Gries and Mukherjee (2010) also use mean pairwise association strength to test multi-unit candidates. Finally, it should be noted that all measures discussed below are implemented in both left-to-right and right-to-left directions, although the discussion is streamlined by exemplifying each measure in a single direction.

The second multi-unit measure is the difference between the mean ΔP with and without the candidate's edge members. In other words, going from left-to-right, this measures the difference between the association between A-B-C-D and B-C-D: Do we gain or lose association by extending the unit? This measures whether the longer version of the candidate increases or decreases the overall association strength. Given that the evaluation is trying to discover the optimum candidates, those candidates which reduce the mean association strength can be viewed as less than optimum. This measure has two variants, one looking at the front and the other at the end of the candidate (and each, like the underlying ΔP, is calculated in both directions), as shown in Table 4.

The third multi-unit association measure is the ΔP of the first unit and the rest of the candidate (A|BCDEFG) and the ΔP of the last unit and the rest of the candidate (ABCDEF|G). This is an alternate measure of how much the increased length raises or lowers the overall association strength. This is calculated as in Table 5 (and, as before, in both directions).

The fourth multi-unit measure uses the dominant pairwise direction of association. In other words, moving through the candidate, is the left-to-right or right-to-left association stronger between the current pair of units? The idea here is that the optimum candidate should have a single dominating direction, and that the more disagreement there is in pairwise directional associations the worse the candidate is. This sort of measure was suggested, for example, by Gries (2013), although not implemented. The assumption that a construction should have a single dominating direction of association is not entirely transparent, and further work needs to be done on this issue.

There are two methods of calculating this measure, a scalar method and a categorical method. First, the scalar method finds the difference between both directions for each pairwise unit and sums these differences. Positive numbers indicate the dominance of left-to-right association while negative numbers indicate dominance of right-to-left association. This provides both the direction and the degree of the dominance. One weakness, however,

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

TABLE 4. *Calculating the Reduced ΔP*

1	Beginning-Reduced ΔP = Mean ΔP(ABCDEFG) – Mean ΔP(BCDEFG)
2	End-Reduced ΔP = Mean ΔP(ABCDEFG) – Mean ΔP(ABCDEF)

is that two large pairwise differences can cancel each other out. Thus, the related categorical measure simply counts the number of pairs for which the left-to-right or right-to-left measure dominates and returns the minimum of these as a counter of how many times the dominating direction changed while moving sequentially through the candidate. Thus, a candidate in which either direction of association wholly predominates would receive a 0, a candidate with one change in direction would receive a 1, and so on. These are calculated as shown in Table 6.

This collection of association measures, together with frequency, is used to create a vector representing each candidate. A summary of the measures contained in this vector is given in Table 7. The selection and ordering of possible candidates is performed using this vector representation. This is, as all quantitative models are, a simplification of a construction grammar, in this case focusing only on frequency and frequency-based co-occurrence information to determine which potential constructions form the strongest or most associated units. The question, however, is whether this simplification (i.e., purely statistical generalization) is sufficient for learning a construction grammar from a corpus.

Alternate methods for calculating multi-unit association strength include Wei and Li (2013), who start with da Silva and Lopes' (1999) notion of pseudo-bigrams, in which all sequences longer than two units are reduced to all possible pairwise combinations (e.g., A|BCD, AB|CD, ABC|D for the sequence ABCD). This is similar to the divided ΔP measures described above. Starting with these pseudo-bigrams, Wei and Li take the average pointwise mutual information score for each pseudo-bigram in the sequence, but refine the average by weighting each pseudo-bigram by its probability in the corpus. This gives more weight in the final measure to the most probable subsequences.

The one assumption that these measures require is that the language learner is able to store frequencies, both of units and of sequences. In other words, a sizable amount of linguistic memory is required to store all the units and sequences that make up possible candidates and to update the frequencies of those units and sequences as new language is observed. This could be done, in algorithmic terms, either with cumulative observed frequencies or with a rolling time-based window. This approach, then, does assume that learners are capable of this sort of frequency storage, a question that is beyond the scope of the present paper (although see Tomasello, 2003, and Bybee, 2010).

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

TABLE 5. *Calculating the Divided ΔP*

1	Beginning-Divided ΔP = ΔP(A | BCDEFG)
2	End-Divided ΔP = ΔP(ABCDEF | G)

For the sake of example, sample calculations are shown for the sequence *did not know about it*. Only lexical items are considered for simplicity. First, this sequence consists of the pairs in (11). Each word is shown with its frequency in the Corpus of Contemporary American English (COCA: Davies, 2010) in brackets, with the total co-occurrences of each pair following. The left-to-right (LR) and right-to-left (RL) ΔP are shown for each (note that the total number of words in COCA is rounded to 520 million in these calculations). Given these measures, the summed ΔP left-to-right is 0.0939 with a smallest pairwise value of 0.0108 ("know about") and the mean ΔP is 0.0234. Going from right-to-left, the summed ΔP is 0.2052 with a smallest pairwise value of 0.0052 ("not know") and a mean ΔP of 0.0513.

(11) a. "did" [895,094] + "not" [2,155,912] and their co-occurrence [128,432]
a'. LR = 0.0581, RL = 0.1395
b. "not" [2,155,912] + "know" [857,571] and their co-occurrence [14,697]
b'. LR = 0.0130, RL = 0.0052
c. "know" [857,571] + "about" [1,444,147] and their co-occurrence [17,933]
c'. LR = 0.0108, RL = 0.0182
d. "about" [1,444,147] + "it" [5,146,411] and their co-occurrence [75,164]
d'. LR = 0.0120, RL = 0.0423

The reduced ΔP compares the mean values for subsequences; the formulation for the beginning-reduced is shown in (12a) and the end-reduced in (12b). For the end-reduced measures, in both directions, the mean association is lower in the longer sequence than in the reduced sequence, although the difference is quite small. The point, though, is to see if a smaller sequence has a higher mean association. It is important to remember that these measures are also calculated on other subsequences if those subsequences are themselves candidates. In this case, for example, each pair is itself a candidate (although not a multi-unit candidate), as are both reduced sequences. This results from the fact that any longer sequence which passes the frequency threshold is composed of subsequences which have also passed the frequency threshold. In practical terms, then, it is the multi-unit measures taken together with the different candidates that allow full coverage in the search for actual constructions and makes iterative measures unnecessary.

TABLE 6. *Calculating the Direction ΔP*

1	Direction-Scalar ΔP = $\Sigma[(\Delta P(A\,|\,B) - \Delta P(B\,|\,A)), (\Delta P(A\,|\,B) - \Delta P(B\,|\,A)) \ldots]$
2	Direction-Categorical ΔP = min(Number LR dominant pairs, Number RL dominant pairs)

(12) a. Beginning-Reduced: Mean (“did not know about it”) – Mean (“not know about it”)
a’. LR = 0.0115, RL = 0.0291
b. End-Reduced: Mean (“did not know about it”) – Mean (“did not know about”)
b’. LR = –0.0039, RL = –0.0030

The divided ΔP calculates multi-unit association with units instead of pairs. This is shown in (13) with its beginning and end variants. The frequency of each unit is shown (in this case, with larger sequences viewed as units), and the frequency of the entire sequence is 16. Longer sequences like this can result in high association: given the sequence *not know about it*, the preceding elements are limited and thus the association is high even though frequency is low. It is important to note, again, that other subsequences are compared in other shorter and longer candidates.

(13) a. Beginning-Divided: (“did” [895,094] | “not know about it” [33])
a’. LR = 0.4831, RL = 0.0000
b. End-Divided: (“did not know about” [197] | “it” [5,146,411])
b’. LR = 0.0000, RL = 0.0714

The final two measures quantify the role of direction within the sequence: Given a series of pairwise associations, how stable is the dominating direction of association? The first measure subtracts the right-to-left association from the left-to-right association in order to show accumulating effects of dominance. In this case, the final measure is –0.1191, showing that, overall, the dominating pairwise direction is right-to-left. The categorical measure looks at how many times the direction changes. In this case, there is one left-to-right dominating pair (“not know”), giving the measure a value of 1. The purpose of this discussion has been to provide an example of how the measures are calculated, rather than a complete analysis of their many permutations.

2.5. MODELING CONSTRUCTIONS

The final and essential step is to take this large number of possible constructions and model the properties which separate possible and actual constructions in order to predict the inventory of the dataset-specific construction grammar. It will be useful, first, to look at some existing approaches to this problem.

TABLE 7. *Summary of measures in vector representing the candidates*

Measure	*Variations*
Simple Frequency	
Summed ΔP	Left-to-Right, Right-to-Left
Mean ΔP,	Left-to-Right, Right-to-Left
Beginning-Reduced ΔP	Left-to-Right, Right-to-Left
End-Reduced ΔP	Left-to-Right, Right-to-Left
Beginning-Divided ΔP	Left-to-Right, Right-to-Left
End-Divided ΔP	Left-to-Right, Right-to-Left
Direction-Scalar ΔP	
Direction-Categorical ΔP	

Wible and Tsao (2010) present StringNet, which finds all sequences of word-form, lemma, or part-of-speech (unigrams to 8-grams) which pass a frequency threshold. StringNet uses a mutual information measure to rank results; however, this measure is not expanded for multi-unit sequences but rather normalized across the results of a particular query. Pruning of nested or redundant sequences is used to reduce the number of candidates. Tsao and Wible (2013) use co-occurrence vectors with these sequences to produce distributional similarity scores. Forsberg et al. (2014) build on StringNet by incorporating dependency parsing to identify phrases as parts of potential constructions, similar to the how the present algorithm reduces complex constituents in identifying potential constructions. Frequency is used to prune potential constructions and the final evaluation is performed using a multivariate generalization of pointwise mutual information (van de Cruys, 2011) scaled by the number of unique word-form sequences instantiating each candidate. Zuidema (2006) formulates the problem of identifying constructions as taking parse trees and identifying those sub-trees which frequently re-occur and which may contain syntactically defined (e.g., partially filled) slots at the end. This approach uses a simpler definition of constructions, along the lines of productive multi-word expressions.

Taken together, this previous work introduces elements present in the current algorithm which are expanded and incorporated into an overall model of a construction grammar in this paper. First, the current algorithm has more robust approaches to dealing with recursive structure (e.g., reducing noun phrases) and partially filled / unfilled slots. Further, it includes semantic category as a level of representation, an important part of representing constructions. These improvements involve the generation of possible constructions. The primary contribution of this paper, however, consists of developing and aggregating measures of association to model the gradient distinction between possible and actual constructions. This component is the essential central problem of construction grammar induction: reducing large

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

numbers of possible representations to a small number of actual and productive constructions. Thus, the current work builds on existing work to produce a coherent and efficient model for construction identification and extraction.

Given a large number of potential constructions with frequency and association strength values, the model for determining which to include in the grammar first removes clear false positives and then ranks the remaining candidates by their degree of entrenchment. The pruning steps, shown in Table 8, begin by removing those candidates which fall below the pairwise threshold. In other words, multi-unit candidates such as ABCDEF have both multi-unit association scores and pairwise scores; the idea here is to remove those candidates which have weak links between at least one pair, indicating that an alternative candidate with alternate boundaries is a better representation.

The second step is to remove those candidates whose mean association strength as a whole is lower than the mean association strength of a subsequence (e.g., ABCDEF vs. BCDEF or ABCDE). The idea here is that the representation with the higher mean association strength is the best grammatical unit.

The third step is to prune those candidates in which the dominating pairwise direction of association changes internally. For example, with the sequence ABCDEF, if all dominating pairwise associations are left-to-right except for CD, in which right-to-left dominates, this is an indicator that the candidate provides a non-optimal boundary.

The final two reduction steps are the simplest: horizontal pruning takes the remaining candidates and chooses the largest, while vertical pruning finds those candidates of the same length which share the same association strengths, so that they are alternate representations of the same underlying construction.

These reduction rules are applied in this order, with association strength given the most weight because it removes the largest number of candidates and thus eases the application of subsequent rules. The final step is to rank the remaining constructions by their degree of entrenchment; in other words, the idea is to order constructions by how highly associated they are. This is done using the mean ΔP and the end-divided and beginning-divided ΔP. First, the highest directional score for each of these three measures is taken, and then again the highest of these scores. Thus, each candidate is represented by its highest direction and type of association measure. In other words, because constructions take many forms and association can be captured by any of these measures, each candidate is represented by its highest association and ranked accordingly.

2.6. CONSTRUCTION IDENTIFICATION AND COLLOSTRUCTIONAL ANALYSIS

The measures of association used to model constructions complement existing work on measuring properties of constructions from corpora. Collostructional

TABLE 8. *From potential to actual constructions*

Order	*Operation*
1	SΔP: Remove candidates which fall below pairwise ΔP threshold
2	RΔP: Remove candidates which lose association strength when reduced
3	Direction: Remove candidates which change directions of association
4	Horizontal Pruning: Keep longest sequence possible within remaining candidates
5	Vertical Pruning: Keep representation with highest association strength

analysis (Stefanowitsch & Gries, 2003, 2005; Gries & Stefanowitsch, 2004a, 2004b) encapsulates the most relevant area of work, performing three related tasks: (i) quantifying the relationship between individual words and a given slot of a given construction; (ii) using the relationship between individual words and a given slot of a given construction to quantify the relationship between similar constructions; and (iii) quantifying the relationship between individual words in two different slots in a given construction. This work differs from the present in that it focuses on quantifying differences within and between constructions while taking the existence of particular constructions as a given. The current work, put in similar terms, focuses on quantifying and modeling the differences between constructions and non-constructions. These non-constructions, like other counter-factuals or ungrammatical forms in linguistic analysis, represent possible alternate generalizations drawn from linguistic expressions. Thus, collostructional analysis looks at variations in the use of constructions, whereas this work looks at variations in inventories of constructions across individuals and speech communities.

2.7. COMPARISON TO EXISTING ALGORITHMS

Knowledge-based approaches to computational linguistics manually build machine-tractable representations of language. Such representations include an ontology of atomic concepts with their properties and connections as well as machine-tractable descriptions of the meaning of linguistic expressions phrased in terms of these atomic concepts (see, for example, Nirenburg & Raskin, 2004; Levison, Lessard, Thomas, & Donald, 2013, and the comparison of these approaches to formal semantics in Dunn, 2015). Both Fluid Construction Grammar (FCG) and Embodied Construction Grammar (ECG) (e.g., Bryant, 2004; Steels, 2004, 2012; Chang, De Beule, & Micelli, 2012) can be viewed as variants of this work, in which hand-crafted but machine-tractable representations of constructions, frames, and concepts are collected and manipulated computationally for various purposes (similar to but expanding on Zadrozny, Szummer, Jarecki, Johnson, & Morhenstern, 1994). These approaches do not interface with natural language (e.g., they do not operate

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

on linguistic expressions). Rather, they should be seen as an extension of introspective analysis of constructions into computational applications by standardizing the units and methods of analysis. These approaches are unable to learn constructions from linguistic expressions and cannot be used to simulate language learning because the representations are themselves a sort of innate representation provided to any algorithms which take them as input.

There are also previous computational treatments of constructions in actual corpora. For example, O'Donnell and Ellis (2010) develop an algorithm for searching a RASP-parsed version of the British National Corpus for instances of two predefined verb–argument constructions. Vincze, Zsibrita, and Istvan (2013) and Istvan and Vincze (2014) computationally distinguish between verb–particle constructions and non-construction verb–particle co-occurrences using a parser to identify candidates and then employing a supervised binary classifier to distinguish those which are part of a construction from those which are not, using lexical, syntactic, and semantic features.

The present algorithm is also an approach to unsupervised grammar induction, the task of learning a generalized grammatical representation from observed language (e.g., from text). Van Zaanen (2000) approaches this task as a problem of finding constituents and their boundaries, so that the task is to identify which units are mutually replaceable. The algorithm compares every pair of sentences, using edit distance to determine which units, if any, are shared by the sentences. Those units which occur with shared structures, then, are constituents which can be mutually replaced. This generates candidate constituents which are then evaluated using the probability that the candidate is a constituent. Dennis (2005) takes a similar approach using part-of-speech sequences rather than word-form sequences and adding a span-based edit distance measure. Clark's (2001) approach to finding clusters of constituent types is to take an input text as a sequence of part-of-speech tags and to cluster sequences of these tags using their distribution. Mutual information (MI: i.e., association strength) is used to filter out redundant or nested candidates, and the MI threshold is determined using minimum description length to evaluate possible grammars (cf. Goldsmith, 2006). Klein and Manning (2002) take yet another approach to finding constituents, starting with all possible subsequences of part-of-speech tags within the same sentence as the candidate set, considering only those candidates which produce binary trees. Given observed sentences and unobserved constituents, Expectation Maximization is used to cluster candidates as actual constituents or non-constituents.

While more current approaches to grammar induction have made a number of improvements (Bod, 2006; Headden, Johnson, & McClosky, 2009; Blunsom & Cohn, 2010; Mareček & Straka, 2013; Spitkovsky, Alshawi, & Jurafsky, 2013), this work has focused on grammar as a tree of dependency relations and on categories with phrase-structure rules, such as in combinatory categorical

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

grammar. The present algorithm, however, focuses on grammar as a set of meaningful and symbolic form–meaning mappings. The output is not a parse tree or a set of categorized dependencies, but rather a mapping between linguistic expressions and schematic constructional representations of those expressions at varying levels of abstraction. Thus, this work is not reviewed in more detail here, although see Heinz, de la Higuera, and van Zaanen (2016) for a general overview of the problem.

3. Evaluating learned grammars

This section presents a rigorous quantitative evaluation of learned grammars. The first part (3.1) describes the general experimental design and provides a qualitative analysis of the sorts of constructions formulated by the algorithm. The next subsection (3.2) begins the quantitative analysis by looking at the distributions of and correlations between the various multi-unit association measures employed. The next part (3.3) examines the grammar's coverage on unseen test sets under different construction pruning conditions. The section after this (3.4) quantifies stability in learned grammars across different sizes of datasets and, after this (3.5), the stability in learned grammars across mutually exclusive datasets, with each instance of the algorithm simulating a single language learner.

3.1. EXPERIMENTAL DESIGN AND QUALITATIVE ANALYSIS OF RESULTS

For the purposes of this evaluation, the construction grammar induction algorithm is run on 1 billion words (40 million sentences) from the ukWac web-crawled corpus of UK domain sites (Baroni et al., 2009). The advantage of using this corpus is, in part, its size. This is important for two reasons: first, it showcases the feasibility of the algorithm in terms of efficiency; second, it allows us to examine the stability of the learned grammar across different subsets of the corpus. Given the grammar learned on this dataset, we start with a qualitative analysis of the sorts of constructions which are included in the grammar, looking at representative examples of constructions identified in the ukWac corpus. Additional constructions and examples are given in the 'Appendix'.

The first example of a learned construction is shown in (14a), with examples in (14b–e). This construction is defined by part-of-speech information and the lemma "be", representing a relative clause with a passive verb. While this generalization covers multiple complementizers and modal verbs, it does not allow for multiple tenses within the verb phrase. It remains, however, a productive and schematic representation that covers a large number of linguistic expressions.

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

(14) a. [Wh-Determiner] + [Modal] + "be" + [Past-Participle]
b. that will be provided
c. that can be played
d. which will be presented
e. that should be made

The second example, in (15a), again consists of parts-of-speech with a single high-frequency lemma, "to". This represents an infinitive verb phrase with an object, which, as shown in (15d), can be generalized to any NP. One weakness with this representation, however, is that the determiner is often part of a noun phrase, so that this representation could be made more general by eliminating the [Determiner] from the construction. Of course, the whole point of a data-driven model such as this is that it builds representations from observed usage and not from intuitions about the most productive schema.

(15) a. "to" + [Verb] + [Determiner] + [Noun]
b. to bring an end
c. to get an idea
d. to use any NP
e. to sell a product

A more item-specific example is shown in (16a), this time including a partially filled slot that is defined only by its semantic category of RELIGION. In this case, the construction reflects the metaphor in which a religious organization takes on the characteristics of a physical body. What separates this as a construction, however, is that whereas literal statements about a body do not require a specific form (*strengthen your body*, *heal your body*, etc.), the interpretation here requires a prepositional phrase in which the type of body is specified (*strengthen the body of the church*, *heal the body of Christ*, etc.). An example of over-identification is shown in (16e), in which *church* is actually referring to a physical object and used as a reference point. Thus, this is not an example of this metaphoric construction, but rather is an over-generalization from the learned representation.

(16) a. [Noun] + [Preposition] + [Determiner] + <RELIGION>
b. body of the church
c. member of the church
d. need in the church
e. west of the church

A simple prepositional phrase construction is shown in (17a), involving spatial relations for a given location. This is a schematic construction that does not differentiate between different spatial relations and different types of locations. This does not, however, preclude the algorithm from learning more specific

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

spatial phrases, which in fact it does. For example, more specific identified constructions include: "in" + NamedEntity; "in" + NP; "through" + NP. These are cases where more item-specific and more schematic constructions overlap.

(17) a. [Preposition] + "the" + <LOCATION>
b. on the site
c. in the area
d. into the city
e. throughout the area

A specific verb phrase construction is shown in (18a), in which a movement verb has an infinitive verb as an object. In this case, the infinitive object shows the purpose of the movement, as in examples (18b–e). The object of the infinitive is not included in this construction, and specifying specific objects would result in a finer-grained analysis.

(18) a. <MOVE> + "to" + [Verb]
b. go to buy
c. come to learn
d. travel to find
e. walk to see

Finally, the example in (19a) shows an identified construction which contains incorrect boundaries. We would expect, given introspective analysis, that some semantic definition of the agent would follow "by", but this is not the case. This illustrates one of the major difficulties of construction grammar induction: modeling a representation abstract enough to cover partially filled slots. In this case, the algorithm fails to find an adequately abstract representation for the agent, and thus a partially filled slot is not posited. The difficulty of finding a sufficiently general partially filled slot on the edges of the construction is that a large number of false positives are possible (e.g., the danger of adding unnecessary generalized slots to many constructions).

(19) a. [Noun] + [Past-Participle] + "by"
b. software developed by
c. information given by
d. article written by
e. training provided by

An important attribute of construction grammars is that fully schematic and fully item-specific representations can co-exist. In other words, an abstract argument structure construction (e.g., the ditransitive) co-exists with separately represented instances of that construction (e.g., the idioms *give me a hand* and *give me a break*). One advantage of this model, then, is that such overlapping constructions of varying abstractness can be captured, so long as each instance

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

itself qualifies as a construction. The point, then, is that this paradigm of grammar induction is not limited a priori to a single level of representation or a single level of abstraction.

A final question here is whether these are posited to be psycholinguistically valid constructions. In other words, are the elements of this grammar supposed to be those present in the mind of a speaker of this language? The goal here is somewhat more indirect: to automatically produce the inventory of constructions necessary to describe the corpus. The question is whether the algorithm can learn adequate grammatical representations from the corpus, not that it necessarily learns exactly the same set as a human in exactly the same manner. This indirectness is a result of the fact that the corpus under study contains language produced by a large number of individuals. If the algorithm were run entirely on a corpus of language produced by a single individual we could consider more direct psycholinguistic tests of the produced grammar. However, a language such as 'English' or even 'British English' is an abstraction over a large number of individuals rather than a representation of the psycholinguistic reality of language in any single individual. Thus, in representing an abstraction in this manner the present algorithm is subject to all the same criticisms as that abstraction in not being specific to the psycholinguistic state of individuals.

3.2. DISTRIBUTIONS OF FEATURE VALUES

The model uses fourteen measures of association for multi-unit potential constructions. Given that these measures are novel implementations for dealing with an open problem, it is important to consider the relative agreement and distributions of these measures. For the evaluation below, the measures are examined across the first 20 million sentences in the corpus, and phrase types (e.g., NP) are not considered, for the sake of simplicity. The descriptive statistics for the measures are calculated using only the subset of sequences which are more than two units in length (a total of 74,522). This is because the multi-unit measures have a zero value for sequences of only two units. Further, no threshold for pairwise association strength is used, unlike for the measures used in the model itself. This is because the threshold effectively gives multi-unit sequences a zero for the summed ΔP score if any pairwise association falls below a set parameter, and this changes the distributions by enlarging the number of zero values. Thus, this evaluation is about comparing the measures on multi-unit sequences without a threshold in order to get a more accurate view of the measures themselves, rather than evaluating the measures as used for reducing candidates in the overall model.

First, the agreement between each of the measures is shown in Figure 2 and Figure 3, calculated using Pearson's R. The question is whether the

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

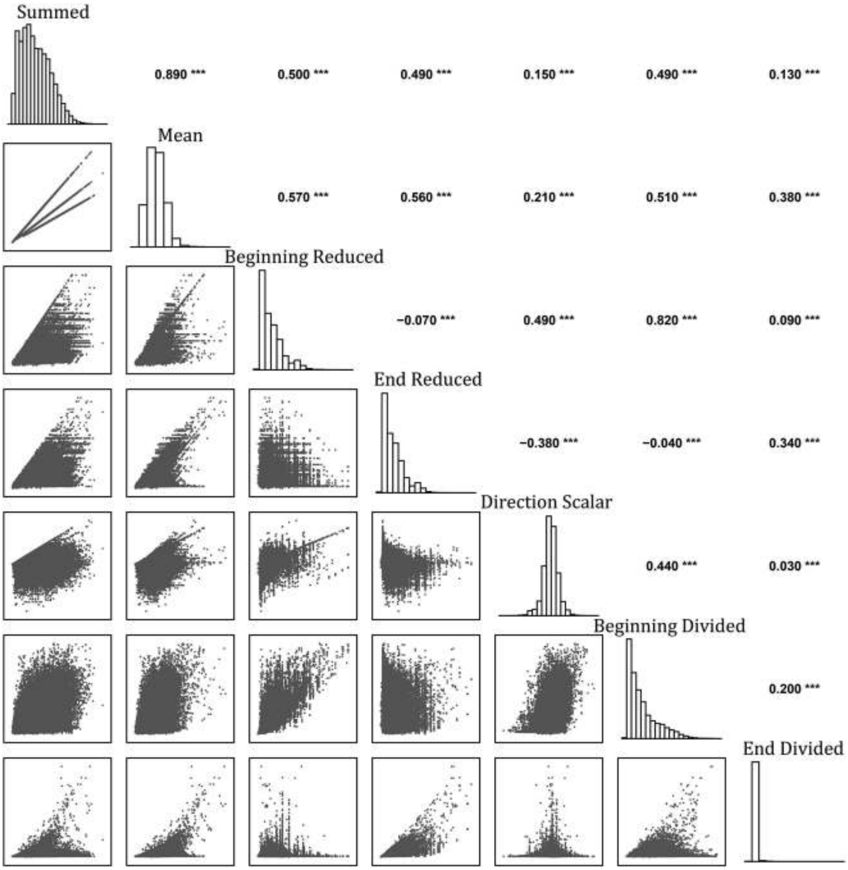

Fig. 2. Left-to-Right Correlations.

measures ultimately represent the same relationships and thus are redundant, or whether they reveal unique aspects of association. These figures show the scatterplots of each pair on the right-hand side, a histogram of each measure's density distribution in the middle, and the correlation coefficient on the left-hand side. Each of the correlations is significant, not surprisingly given the large number of instances.

In both directions the Summed and Mean measures are closely related; the scatterplot shows three distinct degrees of correlation, with the correlation diminishing as the sequences in question grow longer (i.e., the sum and the mean are very similar for shorter sequences, which is expected). Thus, this relationship decreases as candidates grow longer. The two methods for comparing subsequences within a candidate, the Divided and Reduced measures, show little correlation between their respective Beginning and End

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

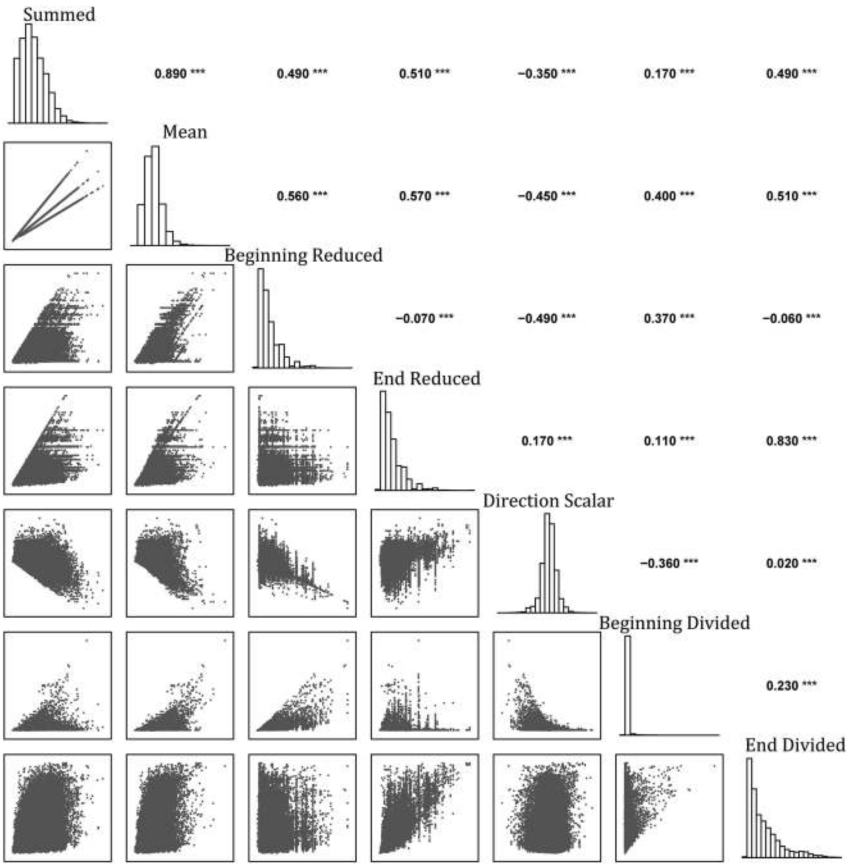

Fig. 3. Right-to-Left Correlations.

variants in both directions (the highest such correlation being 0.230 for the right-to-left Divided measures). The relationship between the Divided and Reduced measures is quite high at the beginning of the sequences (i.e., at the Beginning going left-to-right and at the End going right-to-left), exceeding 0.800 in both cases. However, at the end of the sequences the correlation is much lower (never higher than 0.370). Thus, these variations on the subsequence measure do provide unique information in many but not all situations. For all of these measures, it seems to be the case that they grow less correlated as the sequences in question grow longer. An interesting further question, outside the scope of the present paper, is to what extent sequence length influences the distribution and correlation of association measures, and what alterations can be made to reduce this influence for shorter sequences.

The next question is whether the measures make adequate distinctions between potential multi-unit constructions. We approach this question by looking at measures of the distribution of each of these features, in Table 9, calculated as above across only multi-unit potential candidates in the first 20 million sentences in the corpus. The measures show what we would expect: wide ranges of values with means close to zero. This is because most candidates do not show association. Those which do show internal association are outliers, in a sense, and this is what allows them to be identified as actual constructions. The two measures which do not show means close to zero are the summed values, in both directions. This is a result of the fact that only multi-unit candidates are considered here, so that all instances have at least three units. This, of course, influences the mean value but is necessary to allow this measure to be compared directly with the others.

3.3. DEGREE OF COVERAGE

The ideal construction grammar has at least one construction to account for every linguistic expression in a corpus. In other words, because all linguistic expressions are hypothesized to be formed from an underlying grammatical construction, it should be the case that all attested linguistic expressions can be described by at least one construction in the predicted grammar. Thus, the degree of coverage of a grammar is an important criteria for evaluating a learned construction grammar and, following from this, for evaluating the learning algorithm itself. The measure of coverage is calculated as in (20), in which LE stands for Linguistic Expressions (operationalized in this case as sentences), with *c* standing for the subset covered by a hypothesized construction and *n* for the subset not covered in this way. Thus, this measure is simply the percentage of the test corpus represented by the learned grammar, using sentences as the unit of analysis

(20) $LE_c / LE_c + LE_n$

This evaluation is conducted by applying the grammar learned from the full corpus to an unseen portion of the ukWac corpus in order to determine how much of the unseen corpus is described by the learned grammar. The test set consists of 1.5 million sentences, evaluated in subsets of 100k sentences each, allowing us to evaluate fluctuations in the adequacy of the grammar across different test sets. There is a balance to be reached here between predicting a small set of generalized and highly associated constructions, on the one hand, and predicting a grammar that achieves full coverage on the test sets, on the other. Given this balance, we compare three learned grammars: the 'full pruning grammar' (2,309 constructions) contains only those constructions which pass all the pruning stages discussed above; the 'no pairwise grammar'

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

TABLE 9. *Distribution measures for each feature*

Feature	Mean	Std. Dev.	Range
Frequency	37,527	69,460	12,600–3,681,400
Summed (LR)	0.317	0.188	0.000–1.201
Summed (RL)	0.334	0.204	–0.004–1.544
Mean (LR)	0.105	0.051	0.000–0.524
Mean (RL)	0.112	0.057	–0.002–0.635
Beginning Reduced (LR)	0.105	0.094	–0.016–0.792
Beginning Reduced (RL)	0.110	0.103	–0.018–0.895
End Reduced (LR)	0.106	0.092	–0.016–0.824
End Reduced (RL)	0.111	0.103	–0.018–0.895
Directional Scalar	−0.012	0.152	–1.025–0.946
Beginning Divided (LR)	0.163	0.155	–0.016–0.957
Beginning Divided (RL)	0.006	0.021	–0.005–0.857
End Divided (LR)	0.005	0.016	–0.003–0.601
End Divided (RL)	0.178	0.177	–0.019–0.981

(26,223 constructions) applies the directional and divided ΔP and horizontal pruning stages, but does not eliminate candidates using the pairwise threshold. Finally, the ‘no pruning grammar’ (101,503 constructions) does not apply any of the pruning rules (except, of course, the construction frequency threshold). This allows us to see how expanding the grammar increases the overall coverage on these test sets.

The results are shown in Figure 4, with percentage of coverage across the subsets of the test corpus shown for each grammar. First, the coverage is consistent across both grammars and test sets. In other words, each grammar has very similar coverage across different test sets, showing consistency in the adequacy of the grammar on unseen linguistic expressions. Further, the difference between the models is maintained across test sets. For example, both the third and twelfth sets show a dip in coverage that is observed with all models. This shows that the coverage tests are stable measures of the quality of a grammar’s coverage (regardless of the size or generalizability of the grammar).

The coverage experiment shows that larger grammars (e.g., without pruning) have more coverage. However, this increased coverage is not proportional to the size of the grammar. Thus, the fully reduced grammar is only 2% of the size of the full grammar and yet maintains coverage between 5% and 10% lower than the much larger grammar. Thus, while some important elements of the grammar have been discarded, the association measure model allows a much smaller grammar to find most of the optimum constructions. This is significant because the problem is to maintain high coverage on unseen test sets without simply positing a very large grammar: the small pruned grammar contains few false positives, even if it misses some true positives.

The selection or learning of the grammatical constructions from the total hypothesis space involves a combination of association measures (to model

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

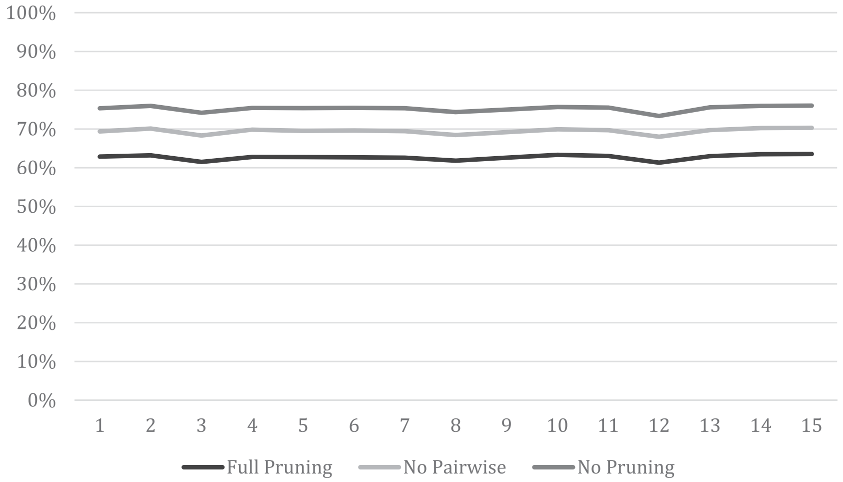

Fig. 4. Degree of coverage across test sets of 100k sentences.

which sequences are more cognitively entrenched than others) and pruning rules (to use those association measures to reduce the number of predicted constructions). We can thus use the coverage experiment to show which association measures were most useful for producing a small grammar with high coverage. With only frequency measures, the grammar consists of 101,503 sequences which could potentially be a grammatical representation; this is reduced to 26,223 sequences with all pruning except the pairwise threshold (e.g., the reduced and directional measures). This is further reduced to 2,309 with the pairwise threshold. While coverage is reduced with each reduction in the grammar, these reductions are minimal. A further examination of the amount of influence of each measure individually (e.g., comparing performance with different subsets of association measures) is beyond the scope of this paper, in large part because such tests would be much more meaningful in a multi-language context: Which measures perform best for which language? The question here is whether these measures can be used to produce a meaningful grammatical representation in the first place.

While the model can always be improved, these coverage results show that observed frequencies can be used to model the productive elements of a grammar and distinguish them from possible but not productive elements. In other words, the frequency threshold has reduced the enormous number of potential constructions to a smaller but still large number of candidates, and the association strength measures have reduced this to a small grammar while maintaining relatively high coverage across sets of unseen linguistic expressions.

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

3.4. STABILITY ACROSS CORPUS SIZES

Given the grammar induction algorithm, how much variation is there in the learned or predicted grammars given the size of the corpus used for evaluation? Another way of looking at this question is how large a corpus needs to be before the algorithm converges onto a stable output grammar. This question is approached by running the algorithm on increasingly large subsets of the corpus and determining, for each subset, how much its grammar agrees with the final grammar. All non-frequency thresholds are held constant across corpus sizes, while the frequency thresholds are scaled relative to the size of the corpus. The results are shown in Table 10, along with the number of constructions in the grammar for each subset (note that the number of constructions in the full grammar here differs from the other evaluations as a result of scaling the frequency thresholds; this scaling was performed in order to reduce the influence of absolute frequency on the results).

Agreement is calculated using precision: given the grammar learned from a subset of the corpus, how many of the identified constructions are present in the full, gold-standard grammar? This measure is quantified as in (21), where FP stands for false positives (those elements in the subset grammar not present in the full grammar) and TP stands for true positives (those elements in both grammars).

(21) $\text{Precision} = \text{TP}_{\text{subset}} / (\text{TP}_{\text{subset}} + \text{FP}_{\text{subset}})$

The results in Table 10 show that stability increases as more data is given to the algorithm. For example, the first sizable increase in agreement is between 10 and 20 million sentences. It is interesting that, even though the subsets have scaled frequency thresholds, the number of candidates decreases as the amount of data increases. This is because the model is more clearly able to separate the grammatical representations from noise as the dataset becomes larger. Given the cap on this experiment, the question of how much data is required for convergence is left open. A further question is whether frequency or association measures have more impact on the amount of data required for convergence. That is a question for further work; the point here is that agreement increases as more data is available, but that convergence is not yet reached.

3.5. STABILITY ACROSS LEARNERS

An argument for innate structure, advanced by Lidz and Williams (2009), is that learners produce very similar grammars for a language even though subject to different observed input. This results, they argue, from innate constraints. Here we turn this into an empirical question: To what degree do instances of the same grammar induction algorithm (i.e., language learners)

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

TABLE 10. *Grammar agreement across corpus sizes*

Corpus Size (Sents)	*Total Constructions*	*Precision*
1 million	2,532	0.2890
5 million	2,167	0.2644
10 million	1,439	0.2966
20 million	1,201	0.3780
40 million	911	n/a

agree in their learned grammars when provided mutually exclusive subsets of the same size? In other words, how much agreement is there when the algorithm is run on different datasets? If the output grammars largely agree, this is evidence that such innate constraints are not, in fact, required to explain this stability in learned grammars. Figure 5 shows the agreement between the grammars produced on four distinct subsets of the corpus, each containing 10 million sentences. Agreement is calculated as the number of shared constructions given the total number of constructions, comparing all subsets to subset 1 for the sake of visualization.

The agreement ranges from the low- to mid-70s. This is quite strong, especially considering the measures of stability by size discussed above (i.e., it would likely be higher if the size of each subset was increased to 20 or 40 million sentences). This means that the algorithm, given entirely different datasets, produced grammars sharing over 70% of their constructions. While by no means perfect, this shows that the grammar induction algorithm is not burdened with a poverty-of-the-stimulus that requires innate structure to produce consistent output across learners. In other words, the hypothesis of innate structure is not required to explain relatively consistent grammars from different language learners.

3.6. FURTHER WORK

As always in projects of this sort, further work is necessary to explore issues raised in the course of these experiments. First, the dependencies should be reduced as much as possible to maintain a fully unsupervised pipeline. This has, in fact, been accomplished with additional algorithms for forming distributional semantic dictionaries and for learning phrase structure rules from a part-of-speech parsed corpus. Such work only strengthens the evidence already presented in this paper. A further important task is to evaluate these and other multi-unit association measures and their influence on the final output construction grammar. Such an evaluation ultimately requires a multi-language and multi-genre experimental design, which renders it outside the scope of the present paper.

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

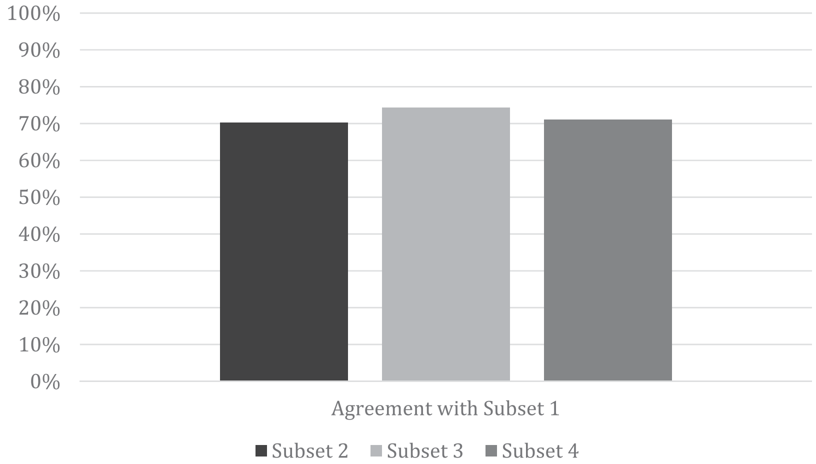

Fig. 5. Stability across simulated learners.

4. Conclusions From evaluations

Grammar induction algorithms, much like language learners, observe very large numbers of linguistic expressions and must generalize from these observations to a relatively small grammar that has the ability to produce all such observations. The problem is that there are a very large number of possible grammatical representations for these observations, unless the space of possible grammatical representations is reduced by positing innate structures/rules/constraints that eliminate many candidates a priori. This paper has shown that the construction grammar induction algorithm presented here can learn a relatively small grammar while (i) maintaining relatively high coverage on unseen linguistic expressions and (ii) maintaining relatively high stability across learners.

The results are by no means perfect and continued technical and theoretical improvements are possible and, in fact, under way. However, these results are sufficient to provide empirical evidence against the poverty-of-the-stimulus line of reasoning for Universal Grammar. This source of evidence, further, is unique in providing large-scale corpus-based evidence for a question which in the past has been approached with small-scale intuition-based evidence. In other words, past work has simply posited that such grammar learning is not possible without constraining innate structures/rules/constraints (e.g., Lidz & Williams, 2009). This paper, on the other hand, goes beyond simple positing and provides empirical evidence that such learning is, in principle, possible.

The question here is whether linguistic structure (specifically, a construction grammar) can be learned from observed language without existing structure or knowledge about the language. In other words, is the grammar wholly

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

learned or is the grammar in part pre-existing? While this algorithm has dependencies (e.g., part-of-speech tagging), this is a practical issue in the sense that data-driven part-of-speech tagging does not need to be reinvented when its current state-of-the-art performs quite well. What this means is that grammatical representations can be learned from observed frequencies. While there are always technical improvements to be made, the current algorithm shows that the learning of grammatical structures in this way is possible and in this sense provides converging evidence with many other empirical sources that have been collected within the Cognitive Linguistics paradigm.

REFERENCES

Baroni, M., Bernardini, S., Ferraresi, A., & Zanchetta, E. (2009). The WaCky Wide Web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, **43**, 209–226.

Blunsom, P., & Cohn, T. (2010). Unsupervised induction of tree substitution grammars for dependency parsing. In H. Li, & L. Màrquez, L. (Eds.), *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1204–1213). Stroudsburg, PA: Association for Computational Linguistics.

Bod, R. (2006). Exemplar-based syntax: how to get productivity from examples. *The Linguistic Review*, **22**, 291–320.

Briscoe, T. (2000). Grammatical acquisition: inductive bias and coevolution of language and the language acquisition device. *Language*, **76**(2), 245–296.

Bryant, J. (2004). Scalable construction-based parsing and semantic analysis. In R. Porzel (Ed.), *Proceedings of the Second International Workshop on Scalable Natural Language Understanding* (HLT-NAACL) (pp. 33–40). Stroudsburg, PA: Association for Computational Linguistics.

Bybee, J. (2006). From usage to grammar: the mind's response to repetition. *Language*, **82**(4), 711–733.

Bybee, J. (2010). *Language, usage, and cognition*. Cambridge: Cambridge University Press.

Chang, N., De Beule, J., & Micelli, V. (2012). Computational construction grammar: comparing ECG and FCG. In L. Steels (Ed.), *Computational issues in Fluid Construction Grammar* (pp. 259–288). Berlin: Springer.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Chomsky, N. (1975). *Logical structure of linguistic theory*. Philadelphia: Springer.

Clark, A. (2001). Unsupervised induction of stochastic context-free grammars using distributional clustering. In W. Daelemans & R. Zajac (Eds.), *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning*. Stroudsburg, PA: Association for Computational Linguistics.

da Silva, J., & Lopes, G. (1999). A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora. In *Proceedings of the Sixth Meeting on the Mathematics of Language* (pp. 369–381). Stroudsburg, PA: Association for Computational Linguistics.

Daudaravičius, V., & Marcinkevičienė, R. (2004). Gravity counts for the boundaries of collocations. *International Journal of Corpus Linguistics*, **9**(2), 321–348.

Davies, M. (2010). The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and Linguistic Computing*, **25**(4), 447–464.

Dennis, S. (2005). An exemplar-based approach to unsupervised parsing. In B. Bara, L. Barsalou, & M. Bucciarelli (Eds.), *Proceedings of the 27th Annual Conference of the Cognitive Science Society* (pp. 583–588). Wheatridge, CO: Cognitive Science Society.

Dunn, J. (2015). Review of Levison, Michael; Lessard, Greg; Thomas, Craig; & Donald, Matthew. 2013. *The Semantic Representation of Natural Language. Studies in Language* **39**(2), 492–500.

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

Fillmore, C. (1988). The mechanisms of 'Construction Grammar.' In S. Axmaker, A. Jaisser, & H. Singmaster (Eds.), *Proceedings of the Fourteenth Annual Meeting of the Berkeley Linguistics Society* (pp. 35–55). Berkeley, CA: Berkeley Linguistics Society.

Firth, J. (1957). *Papers in linguistics, 1934–1951*. Oxford: Oxford University Press.

Forsberg, M., Johansson, R., Bäckström, L., Borin, L., Lyngfelt, B., Olofsson, J., & Prentice, J. (2014). From construction candidates to construction entries: an experiment using semi-automatic methods for identifying constructions in corpora." *Constructions and Frames*, **6**(1), 114–135.

Goldberg, A. (2006). *Constructions at work: the nature of generalization in language*. Oxford: Oxford University Press.

Goldberg, A. (2009). The nature of generalization in language. *Cognitive Linguistics*, **20**(1), 93–127.

Goldberg, A., Casenhiser, D., & Sethuraman, N. (2004). Learning argument structure generalizations. *Cognitive Linguistics*, **15**(3), 289–316.

Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, **27**(2), 153–198.

Goldsmith, J. (2006). An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, **12**(4), 353–371.

Gries, S. (2008). Dispersions and adjusted frequencies in corpora. *International Journal of Corpus Linguistics*, **13**(4), 403–437.

Gries, S. (2012). Frequencies, probabilities, and association measures in usage- / exemplar-based linguistics: some necessary clarifications. *Studies in Language*, **11**(3), 477–510.

Gries, S. (2013). 50-something years of work on collocations: what is or should be next. *International Journal of Corpus Linguistics*, **18**(1), 137–165.

Gries, S., & Mukherjee, J. (2010). Lexical gravity across varieties of English: an ICE-based study of n-grams in Asian Englishes. *International Journal of Corpus Linguistics*, **15**(4), 520–548.

Gries, S., & Stefanowitsch, A. (2004a). Extending collostructional analysis: a corpus-based perspective on 'alternations'. *International Journal of Corpus Linguistics*, **9**(1), 97–129.

Gries, S., & Stefanowitsch, A. (2004b). Co-varying lexemes in the *into*-causative. In M. Achard & S. Kemmer (Eds.), *Language, culture, and mind* (pp. 225–236). Stanford: CSLI.

Headden, W., Johnson, M., & McClosky, D. (2009). Improving unsupervised dependency parsing with richer contexts and smoothing. In M. Ostendorf, M. Collins, S. Narayanan, D. Oard, & L. Vanderwende (Eds.), *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 101–109). Stroudsburg, PA: Association for Computational Linguistics.

Heinz, J., de la Higuera, C., & van Zaanen, M. (2016). *Grammatical inference for computational linguistics*. San Rafael, CA: Morgan & Claypool.

Hilpert, M. (2008). New evidence against the modularity of grammar: constructions, collocations, and speech perception. *Cognitive Linguistics*, **19**(3), 483–503.

Hopper, P. (1987). Emergent grammar. In J. Aske, N. Beery, L. Michaelis, & H. Filip (Eds.), *Proceedings of the Thirteenth Annual Meeting of the Berkeley Linguistics Society* (pp. 139–157). Berkeley, CA: Berkeley Linguistics Society.

Istvan, N., & Vincze, V. (2014). VPCTagger: detecting Verb-Particle constructions with syntax-based methods. In V. Kordoni, M. Egg, A. Savary, E. Wehrli, & S. Evert (Eds.), *Proceedings of the 10th Workshop on Multiword Expressions* (pp. 17–25). Stroudsburg, PA: Association for Computational Linguistics.

Jelinek, F. (1990). Self-organizing language modeling for speech recognition. In A. Waibel & K. Lee (Eds.), *Readings in speech recognition* (pp. 450–506). San Mateo, CA: Morgan Kaufmann.

Katzir, R. (2014). A cognitively plausible model for grammar induction. *Journal of Language Modelling*, **2**(2), 213–248.

Kay, P., & Fillmore, C. (1999). Grammatical constructions and linguistic generalizations: the What's X Doing Y? construction. *Language*, **75**(1), 1–33.

Klein, D., & Manning, C. (2002). A generative constituent-context model for improved grammar induction. In P. Isabelle (Ed.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 128–135). Stroudsburg, PA: Association for Computational Linguistics.

https://doi.org/10.1017/langcog.2016.7 Published online by Cambridge University Press

Langacker, R. (1987). *Foundations of Cognitive Grammar*. Stanford: Stanford University Press.

Langacker, R. (2006). On the continuous debate about discreteness. *Cognitive Linguistics*, **17**(1), 107–151.

Langacker, R. (2008). *Cognitive Grammar: a basic introduction*. Oxford: Oxford University Press.

Levison, M., Lessard, G., Thomas, C., & Donald, M. (2013). *The semantic representation of natural language*. New York: Bloomsbury.

Lidz, J., & Williams, A. (2009). Constructions on holiday. *Cognitive Linguistics*, **20**(1), 177–189.

Mareček, D., & Straka, M. (2013). Stop-probability estimates computed on a large corpus improve unsupervised dependency parsing. In H. Schuetze (Ed.), *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (pp. 281–290). Stroudsburg, PA: Association for Computational Linguistics.

Nirenburg, S., & Raskin, V. (2004). *Ontological semantics*. Cambridge, MA: MIT Press.

Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kubler, S., Marinov, S., & Marsi, E. (2007). MaltParser: a language-independent system for data-driven dependency parsing. *Natural Language Engineering*, **13**(2), 95–135.

O'Donnell, M., & Ellis, N. (2010). Towards an inventory of English verb argument constructions. In M. Sahlgren & O. Knutsson (Eds.), *Proceedings of the Workshop on Extracting and Using Constructions in Computational Linguistics* (pp. 9–16). Stroudsburg, PA: Association for Computational Linguistics.

Piao, S., Bianchi, F., Dayrell, C., D'Egidio, A., & Rayson, P. (2015). Development of the multilingual semantic annotation system. In R. Mihalcea (Ed.), *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1268–1274). Stroudsburg, PA: Association for Computational Linguistics.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

Solan, Z., Horn, D., Ruppin, E., & Edelman, S. (2005). Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences*, **102**(33), 11629–11634.

Spitkovsky, V., Alshawi, H., & Jurafsky, D. (2013). Breaking out of local optima with count transforms and model recombination: a study in grammar induction. In T. Baldwin & A. Korhonen (Eds.), *Proceedings of 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1983–1995). Stroudsburg, PA: Association for Computational Linguistics.

Steels, L. (2004). Constructivist development of grounded construction grammar. In D. Scott (Ed.), *Proceedings of the 42nd Meeting of the Association for Computational Linguistics* (pp. 9–16). Stroudsburg, PA: Association for Computational Linguistics.

Steels, L. (2012). Design methods for fluid construction grammar. In L. Steels (Ed), *Computational issues in Fluid Construction Grammar* (pp. 3–36). Berlin: Springer.

Stefanowitsch, A., & Gries, S. (2003). Collostructions: investigating the interaction between words and constructions. *International Journal of Corpus Linguistics*, **8**(2), 209–243.

Stefanowitsch, A., & Gries, S. (2005). Covarying lexemes. *Corpus Linguistics and Linguistic Theory*, **1**(1), 1–43.

Tomasello, M. (2003). *Constructing a language*. Cambridge, MA: Harvard University Press.

Tsao, N., & Wible, D. (2013). Word similarity using constructions as contextual features. In I. Dagan et al. (Eds.), *Proceedings of the Joint Symposium on Semantic Processing: Textual Inference and Structures in Corpora* (pp. 51–59). Stroudsburg, PA: Association for Computational Linguistics.

van de Cruys, T. (2011). Two multivariate generalizations of pointwise mutual information. In C. Biemann & E. Giesbrecht (Eds.), *Proceedings of the Workshop on Distributional Semantics and Compositionality* (pp. 16–20). Stroudsburg, PA: Association for Computational Linguistics.

van Zaanen, M. (2000). ABL: alignment-based learning. In M. Kay (Ed.), *Proceedings of the 18th International Conference on Computational Linguistics* (pp. 961–967). San Francisco, CA: Morgan Kaufmann Publishers.

Vincze, V., Zsibrita, J., & Istvan, N. (2013). Dependency parsing for identifying Hungarian light-verb constructions. In H. Chen (Ed.), *Proceedings of the International Joint Conference on Natural Language Processing* (pp. 207–215). Asian Federation of Natural Language Processing.

Wei, N., & Li, J. (2013). A new computing method for extracting contiguous phraseological sequences from academic text corpora. *International Journal of Corpus Linguistics*, **18**(4), 506–535.

Wible, D., & Taso, N. (2010). StringNet as a computational resource for discovering and investigating linguistic constructions. In M. Sahlgren & O. Knutsson (Eds.), *Proceedings of the Workshop on Extracting and Using Constructions in Computational Linguistics* (pp. 25–31). Stroudsburg, PA: Association for Computational Linguistics.

Zadrozny, W., Szummer, M., Jarecki, S., Johnson, D., & Morhenstern, L. (1994). NL understanding with a grammar of constructions. In M. Nagao (Ed.), *Proceedings of the International Conference on Computational Linguistics* (pp. 1289–1293). International Conference on Computational Linguistics.

Zuidema, W. (2006). What are the productive units of natural language grammar? A DOP approach to the automatic identification of constructions. In *Proceedings of the 10th Conference on Computational Natural Language Learning*, 29–36.

APPENDIX

Further examples

Construction: [Singular-Noun] + <social act/state> + [Verb] + [Past-Participle]

Examples:
limit people are granted
approach should be used
option should be included
team should be asked
assessment should be kept
program must be recommended
notice must be given
bar should be pressed
NP should be accepted
information should be published

Construction: [Singular-Noun] + [Preposition] + [Number] + <time>

Examples:
delivery within 2 weeks
train within one hour
format within one year
module over six months
increase over ten years
target within three years
mark within six months
change over five years
notice within 7 days

Construction: "be" + [Past-Participle] + "out"

Examples:
was grown out
was sent out
was carried out
was made out
was taken out
was worked out
was given out
was forced out
was set out
was delivered out

Construction: <MOVEMENT> + NP + <TIME>

Examples: here NP time
put NP time
train NP day
set NP time
come NP year
go NP night
course NP day
through NP now
stay NP year
follow NP day

Construction: [Comparative-Adj] + [Singular-Noun]

Examples: further information
more power
great power
more variety
great effort
new knowledge
good standard
large area
high quality
long life

Construction: [Singular-Noun] + <MONEY>

Examples: purchase price
NP price
building costs
housing prices
housing market
energy bill
government fund
development company
family business
capital investment