A Method for Learning Large-Scale Computational Construction Grammars from Semantically Annotated Corpora

Paul Van Eecke*
Artificial Intelligence Laboratory
Vrije Universiteit Brussel
Pleinlaan 2, B-1050 Brussels
paul@ai.vub.ac.be

Katrien Beuls*
Faculté d'informatique
Université de Namur
Rue Grandgagnage 21, B-5000 Namur
katrien.beuls@unamur.be

Abstract

We present a method for learning large-scale, broad-coverage construction grammars from corpora of language use. Starting from utterances annotated with constituency structure and semantic frames, the method facilitates the learning of human-interpretable computational construction grammars that capture the intricate relationship between syntactic structures and the semantic relations they express. The resulting grammars consist of networks of tens of thousands of constructions formalised within the Fluid Construction Grammar framework. Not only do these grammars support the frame-semantic analysis of open-domain text, they also house a trove of information about the syntactico-semantic usage patterns present in the data they were learnt from. The method and learnt grammars contribute to the scaling of usage-based, constructionist approaches to language, as they corroborate the scalability of a number of fundamental construction grammar conjectures while also providing a practical instrument for the constructionist study of English argument structure in broad-coverage corpora.

1 Introduction

A family of linguistic theories called *constructionist approaches to language* (Goldberg, 2003) has established itself over the last four decades as a prominent alternative to mainstream generative linguistics. Constructionist approaches to language build on a number of shared foundational principles, of which the most consequential are that they model all linguistic knowledge needed for language comprehension and production in terms of networks of form-function mappings called constructions, that these networks of constructions are personal, dynamic and learnt through language use, that there exists no strict distinction between the lexicon and the grammar of a language, and that all attested linguistic phenomena are considered to be of equal interest to linguistic analysis (see a.o. Fillmore, 1988; Goldberg, 1995; Kay and Fillmore, 1999; Croft, 2001; Goldberg and Jackendoff, 2004; Langacker, 2008; Hilpert, 2014; Hoffmann, 2022).

Advocates of constructionist approaches to language praise (i) the ability of construction grammars to account for the regular as well as the idiomatic and less-compositional aspects of language (Fillmore et al., 1988; Goldberg, 1995; Kay and Fillmore, 1999; Fried and Östman, 2004; Michaelis, 2012; Fried and Nikiforidou, 2025), (ii) their psychological validity (Bencini and Goldberg, 2000; Tomasello, 2003; Alishahi and Stevenson, 2008; Diessel, 2015), and (iii) the central place that they attribute to the meaning and communicative function of linguistic entities (Langacker, 2003; Fried, 2015). At the same time, a long-standing criticism of constructionist approaches to language is that the scalability of "*precise and testable models [of construction grammar]*" (Bod, 2009) very much remains an open challenge (van Trijp et al., 2022; Boas, 2025; Doumen et al., 2025).

This paper contributes to the scaling of usage-based, constructionist approaches to language by introducing a method for learning large-scale, broad-coverage construction grammars from corpora of language use. Given a corpus of utterances annotated with syntactic structure and semantic frames, the method facilitates the learning of a network of constructions that captures at scale the relationship, whether compositional or not, between semantic frames and the syntactic means that express them. Apart from introducing the method, we also demonstrate its application to a multi-genre corpus of English texts that were systematically and exhaustively annotated with PropBank frames (Palmer et al., 2005). The resulting grammar networks consist of tens of thousands

*Both authors contributed equally to this paper and declare that it was conceived and written without assistance from generative AI tools.

of human-interpretable constructions that do not only support the frame-semantic analysis of English open-domain text, but also house a trove of information about the syntactico-semantic usage patterns present in the data they were learnt from.

The method has been fully integrated into the open-source Fluid Construction Grammar framework (Steels, 2004; Beuls and Van Eecke, 2023). Along with this paper and its source code, we release a series of pre-trained grammars for English and a collection of PyFCG-compatible scripts (Van Eecke and Beuls, 2025) for processing new corpus data using pre-trained grammars, for learning new grammars from semantically annotated corpora, and for linguistically analysing learnt grammars[1].

The research reported on in this paper corroborates the scalability of a number of fundamental construction grammar conjectures. First and foremost, it confirms the feasibility of learning large-scale, broad-coverage construction grammars from corpora of language use. It provides a scalable operationalisation of the bootstrapping of human-interpretable constructions along with construction-specific grammatical categories, and confirms that large-scale grammars can be operationalised without formally distinguishing between substantive and non-substantive constructions. Finally, it revisits the close relationship between constructions and frames and shows that a broad coverage of (at least) English argument structure can be achieved through shallow generalisations alone.

2 Methodology

The input to the grammar learning process consists in utterances that have been annotated with both their syntactic and their semantic structure. The specific instantiation of the method discussed in this paper was developed for English and starts from constituency analyses on the syntactic side and PropBank rolesets on the semantic side. The choice for constituency structures and PropBank rolesets was made following both theoretical and pragmatic considerations. Theoretically, the choice for a frame-semantic representation subscribes to a rich tradition in both cognitive linguistics (Fillmore, 1976; Langacker, 1987; Baker et al., 1998) and artificial intelligence (Minsky, 1974; Schank and Abelson, 1977). The choice for PropBank rolesets specifically was made based on the availability of substantial corpora with systematic and exhaustive annotations, in particular OntoNotes (Weischedel et al., 2013) and English Web Treebank (EWT) (Bies et al., 2012). The choice for constituency structures was made considering that each role in a PropBank annotation always covers the lexical material held by a constituent and that accurate and reliable constituency parsers, such as the Berkeley Neural Parser (Kitaev and Klein, 2018), are available for English.

A schematic example of a single annotated utterance from the training corpus is shown in Figure 1. The left side of the figure shows the utterance "*Old Li Jingtang still tells visitors old war stories.*" (`OntoNotes CCTV`) along with its syntactic structure as obtained using spaCy's implementation of the Berkeley Neural Parser[2] (somewhat condensed and simplified for illustrative purposes). The top-right part of the figure shows the utterance's frame-semantic structure as annotated in the corpus. The utterance contains a single instance of the `tell.01` roleset ('*pass along information*'). In this instance, "*tells*" takes up the role of frame-evoking element (`v`) and "*Old Li Jingtang*", "*old war stories*" and "*visitors*" respectively take up the core roles of `arg0` ('*speaker*'), `arg1` ('*utterance*') and `arg2` ('*hearer*'). The adverb "*still*" was annotated as a temporal modifier (`argm-tmp`) to the roleset instance. The definition of PropBank's `tell.01` roleset is shown in the bottom-right part of the image for reference. Given the focus of this paper on argument structure, we will exclusively focus on the core roles of PropBank roleset instances, i.e. roles that are described in the PropBank frame repository[3] and thereby end in a number.

Let us now take the perspective of the learning algorithm, of which the ultimate goal is to learn a construction grammar that captures the relationship between the form and the meaning of English utterances. On the highest level, the algorithm goes through the training corpus once and proceeds utterance by utterance and roleset instance by roleset instance. For each roleset instance, and we will further follow the example introduced in Figure 1, it looks up the constituent node in the syntactic annotation that corresponds to the frame-evoking element in the semantic annotation. In this case, the algorithm establishes that the frame-evoking `v` role is taken up by a node named 'unit-6'. In the same way, it retrieves the correspondence between the

[1] See `https://fcg-net.org/fcg-propbank`.

[2] See `https://spacy.io`.

[3] See `https://propbank.github.io/v3.4.0/frames/`.

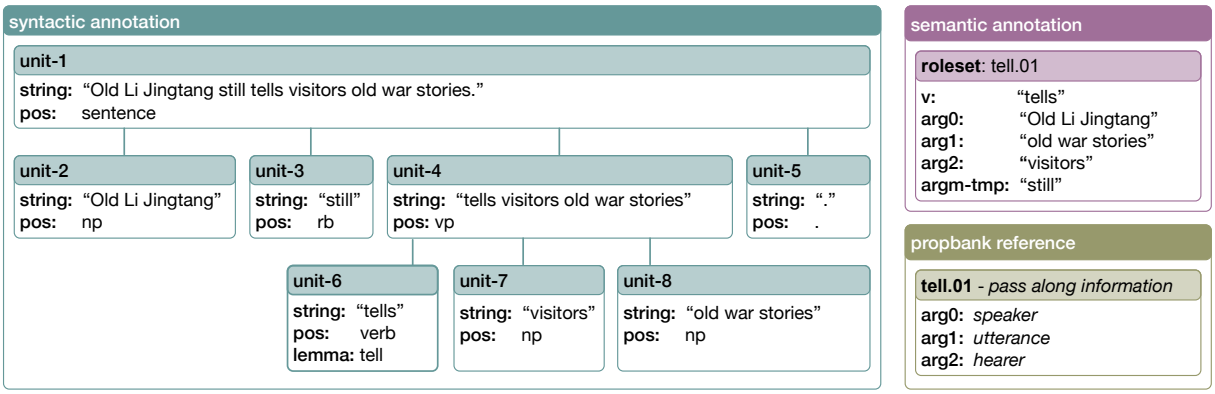

Figure 1: Schematic illustration of the input to the learning process, which consists for each utterance of a syntactic annotation in terms of constituency structure (left) and a semantic annotation in terms PropBank rolesets (right).

`arg0` role and 'unit-2', the `arg1` role and 'unit-8', and the `arg2` role and 'unit-7'. The algorithm then builds three new constructions that are connected through categorial links:

Frame-evoking construction and category The first construction that is built will serve the purpose of recognising potential frame-evoking elements in a constituency structure. Technically, the construction will look for any node with the same lemma and part-of-speech (pos) tag as those held by the syntactic node that was identified to take up the `v` role in the roleset instance being processed. This is shown on the right side of Figure 2, where the feature-value pairs 'lemma: tell' and 'pos: verb' are taken over from 'unit-6' in the input structure. When this construction identifies a node that matches these feature-value pairs, it will contribute the two feature-value pairs shown on the left side of the figure to this node in the structure. The 'roleset' feature, which at this point takes an unbound variable as its value, indicates the frame-evoking potential of the node. The 'fe-cat' feature (short for *'category of frame-evoking element'*) introduces a new category that is proper to this construction and that is thereby indicative of the occurrence of this specific combination of lemma and part-of-speech tag. The value of the 'fe-cat' feature is mnemotechnically named after the observed combination of lemma and part-of-speech tag, in this case 'tell(verb)'. It might seem somewhat counter-intuitive that the construction does not bind the value of the 'roleset' feature to the annotated `tell.01` roleset label. The reason is that the frame-evoking construction should merely identify occurrences of the lemma tell as a verb and indicate their frame-evoking potential, while

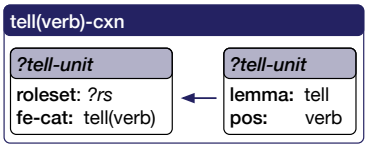

Figure 2: Schematic illustration of the frame-evoking `tell(verb)-cxn` learnt from the input structures shown in Figure 1. This construction identifies uses of the verb 'tell' and indicates their frame-evoking potential.

making at this point no commitment towards specific rolesets they might evoke. For example, the construction that was built here might later be used to indicate the frame-evoking potential of the verb 'tell' in utterances such as "*I could not tell them apart.*" where the `tell.02` roleset is evoked rather than the `tell.01` roleset. Note that logical variables in constructions are indicated by a preceding question mark and that unit names in constructions are always logical variables. Indeed, it would make no sense to refer directly to the names of nodes in the input structure, as these do not carry any meaning apart form providing unique identifiers to nodes.

Argument structure construction and category The second construction that is built will serve the purpose of linking specific nodes in a syntactic structure to the core roles (i.e. roles ending in a number) that these nodes take up in a roleset instance. Technically, the construction will look for a specific constellation of nodes in a syntactic structure that shares certain structural properties with the syntactic annotation in the training example. These structural aspects concern first of all the part-of-speech tags of the nodes that were identified to take up a core role in the annotated roleset instance that is being processed. This is illustrated in Fig-

ure 3, in which the units '*?arg0-unit*', '*?arg1-unit*' and '*?arg2-unit*' each hold the feature-value pair 'pos: np' as taken over from 'unit-2', 'unit-8' and 'unit-7' in the input structure respectively. Then, a unit is added that represents the node that was identified as taking up the 'v' role in the roleset instance. This unit does not incorporate a specific part-of-speech tag, but matches on an 'fe-cat' feature that should previously have been contributed to the structure by a frame-evoking construction (such as the `tell(verb)-cxn` above). Crucially, the value of this feature is not the same as the one that was contributed by this frame-evoking construction. Instead, a new category is introduced that is unique to the argument structure construction being built, along with a link between the categories of the frame-evoking construction and the argument structure construction in the construction network. In this way, the argument structure construction is not tied to a particular frame or roleset, but merely captures the correspondence between structural aspects of the syntactic and semantic representations that are used. In our example, the 'fe-cat' matched by the '*?v-unit*' in the argument structure construction is mnemotechnically called 'arg0(np)-v(v)-arg2(np)-arg1(np)-1' after the syntactico-semantic argument structure pattern that the construction captures. The link between 'arg0(np)-v(v)-arg2(np)-arg1(np)-1' and 'tell(verb)' that is added to the construction network indicates the *compatibility* of both constructions, or in other terms, that this particular ditransitive construction is likely to be a good candidate for assigning a semantic role layout for frames evoked by the verb 'tell'. A final structural element that is incorporated into the construction concerns the *syntactic pathways* through which the core role units are connected with the frame-evoking unit. Each syntactic pathway takes the form of a chain of units solely represented by means of their part-of-speech tag, along with the information of whether the next unit in the chain is either a parent or a child of the previous unit. For example, the '*?arg0-unit*' is connected to the '*?v-unit*' up through a 'pos: sentence' unit and then down through a 'pos: vp' unit. Likewise, the '*?arg1-unit*' and the '*?arg2-unit*' are connected to the '*?v-unit*' up through a 'pos: vp' unit and then down to the frame-evoking unit. Note that while the visualisation of the construction in the figure lays out this constellation of units and pathways in the form of a tree, word order constraints are only incorporated where the pathways

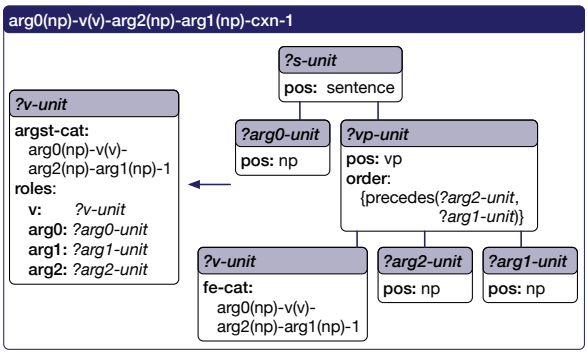

Figure 3: Schematic illustration of the argument structure construction `arg0(np)-v(v)-arg2(np)-arg1(np)-cxn-1` learnt from the structures in Figure 1. This construction identifies a particular type of ditransitive utterance and labels the constituents that take up the roles of `v`, `arg0`, `arg1` and `arg2`.

connecting core role units with the frame-evoking unit would otherwise be indistinguishable. In the case of our ditransitive construction, this is the case for the pathways between the frame-evoking unit on the one hand and the '*?arg1-unit*' and '*?arg2-unit*' on the other. This is detected by the algorithm and a precedence constraint between the two units is therefore incorporated into the construction, stating that it is only applicable if the 'np' representing the `arg2` precedes the 'np' representing the `arg1`.

If this syntactic constellation matches the syntactic structure of a given input utterance, the construction will contribute two new top-level features to the frame-evoking unit in the structure. The 'argst-cat' feature holds the category proper to this argument structure construction, which was also used for matching the 'fe-cat' of the '*?v-unit*' through the construction network. The 'roles' feature lists for each core role in the roleset projected by the construction the unit that represents the constituent that spans the part of the utterance that takes up this role. For example, the value of the 'arg2' subfeature of the 'roles' feature is the variable '*?arg2-unit*', which is bound to the unit that can be syntactically identified by starting at the frame-evoking element, going up to a 'pos: vp' unit and down to a 'pos: np' unit that precedes another 'pos: np' unit. Note that this construction projects a roleset layout without committing to a particular roleset label (such as `tell.01`), and without incorporating a particular frame-evoking element (such as 'tell(verb)'). The algorithm has at this point only learnt that this ditransitive argument structure construction is compatible with occurrences of the

verb 'tell' and represents this knowledge through a link to the `tell(verb)-cxn` in the construction network. Later however, it might learn to link this construction to other frame-evoking constructions and reuse it for utterances that feature the same syntactico-semantic correspondence but use different frame-evoking elements, such as "*Let me give you an introduction*" (`OntoNotes CCTV`).

Roleset construction and category The last construction that is built will serve the purpose of attributing a specific roleset label to a roleset instance that was constructed by the combination of a frame-evoking construction and an argument structure construction. As illustrated in Figure 4, such a construction will look for a unit that holds at least two features: an 'fe-cat' feature contributed by a frame-evoking construction and an 'argst-cat' feature contributed by an argument structure construction. A new category that is proper to the roleset construction is created, and both features take this category as their value. At the same time, two compatibility links are added to the construction network. The first link connects the category of the roleset construction ('tell.01') to the category of the frame-evoking construction ('tell(verb)') and the second link connects the category of the roleset construction ('tell.01') to the category of the argument structure construction ('arg0(np)-v(v)-arg2(np)-arg1(np)-1'). When the roleset construction identifies a unit of which both the 'fe-cat' and the 'argst-cat' are compatible, it will contribute the `tell.01` roleset label as the value of the 'roleset' feature in that unit. The choice for attributing a particular roleset is thus determined by both the frame-evoking element and the argument structure in which it appears. When more links are later added to the grammar network as more training examples are processed, the same `tell.01` construction can be reused in combination with other frame-evoking or argument structure constructions to process utterances such as "*Ah, why should you come out if you were told not to?*" (`OntoNotes CCTV`).

Let us now have a closer look at how the constructions that were learnt can readily be used to process the previously unseen yet constructionally similar utterance "*First, Moses told the people every command in the law.*" (`OntoNotes NT`). First of all, the spaCy-BeNePar parser is called to provide a syntactic analysis of the utterance. This analysis is shown in a condensed

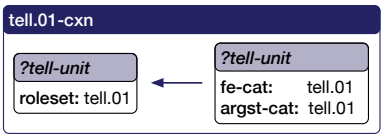

Figure 4: Schematic illustration of the roleset construction `tell.01` learnt from the structures in Figure 1. This construction attributes the `tell.01` roleset to roleset instances identified by connected frame-evoking and argument structure constructions.

form under step ① in Figure 5. Then, Fluid Construction Grammar's construction application process starts. The `tell(verb)-cxn` detects that 'unit-7' matches the features 'lemma: tell' and 'pos: verb', indicates its frame-evoking potential, and contributes the frame-evoking category 'tell(verb)' to the unit (step ②). After that, the `arg0(np)-v(v)-arg2(np)-arg1(np)-cxn-1` is able to match its unit schema with the current structure, under the condition that its '*?v-unit*', '*?s-unit*', '*?arg0-unit*', '*?vp-unit*', '*?arg2-unit*' and '*?arg1-unit*' are respectively bound to 'unit-7', 'unit-1', 'unit-4', 'unit-5', 'unit-8' and 'unit-9' in the input structure, and that its 'arg0(np)-v(v)-arg2(np)-arg1(np)-1' category is linked to the 'tell(verb)' category through the construction network. The blue line that connects both constructions indicates that this condition is fulfilled. The construction now contributes its 'argst-cat' and explicitly lists 'unit-7', 'unit-4', 'unit-9' and 'unit-8' as respectively taking up the roles of `v`, `arg0`, `arg1` and `arg2` in the roleset instance identified by the `tell(verb)-cxn` (step ③). The `tell.01-cxn` matches its 'tell.01' category through the 'fe-cat' feature with the category contributed by the `tell(verb)-cxn` and through the 'argst-cat' feature with the category contributed by the argument structure construction. The three constructions are indeed linked in a triangular way in the construction network, as indicated by the dark blue lines. The `tell.01-cxn` then binds the value of the 'roleset' feature to `tell.01` (step ④). Finally, the roleset instance is extracted from the structure, formalising that a '*pass along information*' event took place, in which "*Moses*" took up the role of '*speaker*', "*every command in the law*" the role of '*utterance*' and "*the people*" the role of '*hearer*' (step ⑤). For a discussion of how such constructions can be used by a hybrid symbolic/distributional processing engine, we refer the interested reader to Verheyen et al. (2025).

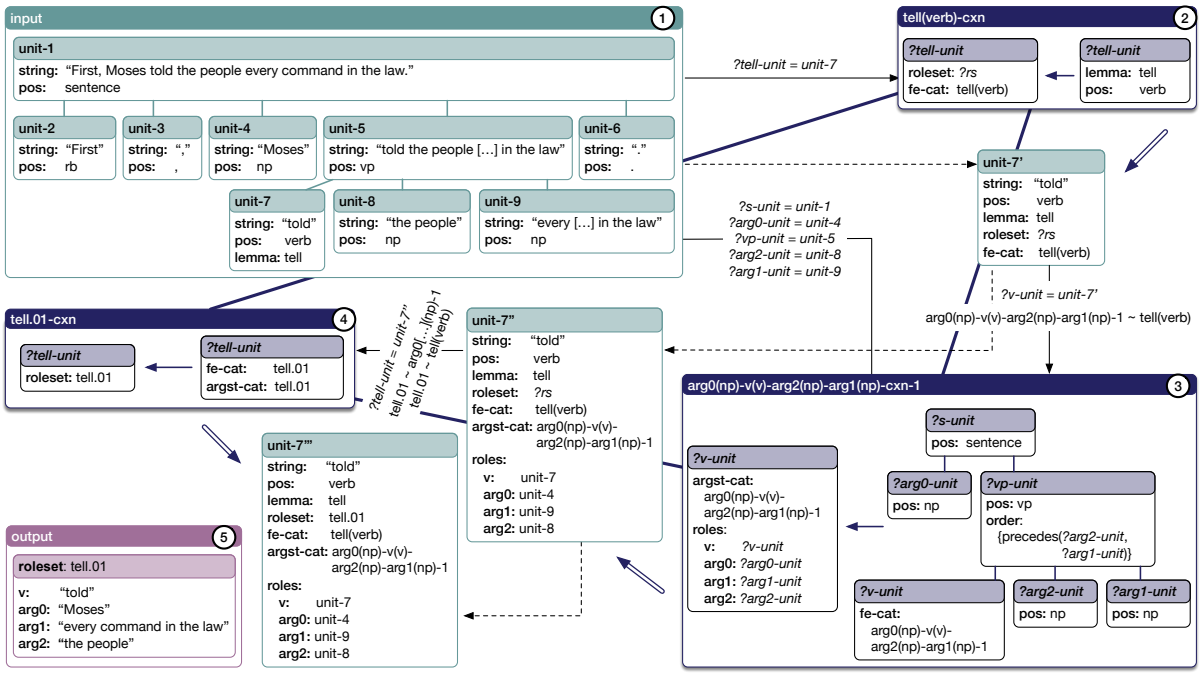

Figure 5: Schematic illustration of the frame-evoking `tell(verb)-cxn`, the ditransitive `arg0(np)-v(v)-arg2(np)-arg1(np)-cxn-1` and the roleset-attributing `tell.01-cxn` combining to process the previously unseen utterance "*First, Moses told the people every command in the law.*". The blue lines connecting the three constructions symbolise their interconnection in the construction network.

The learning algorithm follows the same procedure for learning constructions for each roleset annotated in each utterance of a given corpus. Whenever a construction would be built that already exists in the construction network (up to renamings of variables and categories), a link to the existing construction is added to the construction network rather than a duplicate construction. The frequency of occurrence of both constructions and links is tracked in the construction network during training. The construction network captures thus exactly how frequent its different constructions are, as well as how frequently they combine with each other.

3 Learning a Grammar for English

We apply the methodology to the combination of two English corpora for which high-quality PropBank annotations are available: OntoNotes (Weischedel et al., 2013) and the English Web Treebank (EWT) (Bies et al., 2012). The PropBank-annotated OntoNotes corpus comprises various genres of text ranging from news and talk shows to weblogs and conversational telephone speech. The English part of the corpus consists of 137,812 utterances annotated with 390,266 roleset instances. The PropBank-annotated EWT corpus (Pradhan et al., 2022) also comprises multiple genres of text, including weblogs, newsgroups, email, reviews, and question-answer pairs. It consists of 16,579 English utterances annotated with 50,562 roleset instances[4]. Together, the combined corpus contains 154,391 utterances, annotated with a total of 440,528 roleset instances.

The resulting grammar consists of a network of 40,688 constructions, of which 9,800 are frame-evoking constructions, 22,568 are argument structure constructions and 8,320 are roleset constructions. When it comes to their frequency of occurrence, the constructions in the network follow a Zipfian distribution, i.e. the same distribution as the one observed for the lexical items of a language (Zipf, 1936, 1949; Piantadosi, 2014). This means that the frequency of occurrence of a construction is approximately inversely proportional to the rank of the construction in a table in which all constructions are sorted by decreasing frequency. Interestingly, this observation also holds for each group of constructions in isolation, including the group of argument structure constructions, which are not tied to specific lexical items or other substantive material. The Zipfian distribution of the constructions in the network is visually shown in Figure 6. The left graph plots the frequency of the

[4]See the LDC2013T19 and LDC2012T13 documentation.

constructions in function of their rank on a log-log scale, where the near-straight lines are indicative of a power law relationship between the frequency of constructions and their rank. The right graph displays the frequency of the 50 most frequent constructions on a linear scale, clearly showing their short-head, long-tail distribution. Just like it has been observed that about half of the words in large linguistic corpora are hapax legomena (Malmkjaer, 2004; Kornai, 2008), we observe that almost half of the constructions of the network (48.4%) occur only once in the corpus (see Appendix A for more detailed frequency counts).

The constructions of the grammar network are interconnected through categorial links, of which the weights indicate how often each combination of constructions has been observed in the training corpus. The network thereby holds a trove of empirical information about the frequency of particular syntactico-semantic usage patterns. Let us consider for instance the ditransitive double object construction. We can straightforwardly query the grammar network for those rolesets that are most typically associated with this construction (`give.01` (650), `tell.01` (128), `show.01` (62), ..., `win.01` (1)) and indeed observe that `explain.01` is not in this list (cf. Goldberg, 2019). Interestingly, while `tell.01` ('*pass along information*') integrates frequently with the double object construction, the `tell.02` roleset ('*distinguish, determine*') never does. The network reveals that this roleset is instead strongly associated with constructions that syntactically express its `arg1` role by means of a subclause ("*couldn't tell whether...*", "*can tell that ...*"). This could be taken as evidence that argument structure constructions substantially influence the lexical meaning of verbs, in our example in particular that the 'transfer' aspect in the meaning of the `tell.01` is contributed by the ditransitive construction that it combines with rather than by the verb 'tell' itself. A different question to which the grammar network can provide a straightforward answer concerns the similarity of constructions in terms of their frequency of combination with other constructions. For example, the frame-evoking constructions in the network that are closest to the `tell(verb)-cxn` in terms of their co-occurrence with the argument structure constructions of the grammar are the `ask(verb)-cxn`, the `remind(verb)-cxn` and the `teach(verb)-cxn`, while the `swim(verb)-cxn` and the `consist(verb)-cxn` are among the most distant ones[5].

	Precision	Recall	F_1 score
Roleset	76.15	76.36	76.25
Frame	79.85	80.07	79.96

Table 1: Performance of a 'pure' grammar on the task of semantic frame extraction (1,000 held-out utterances).

While the primary goal of the method is to provide a deeper insight into the language, the fact that the learnt grammars are fully formalised within the Fluid Construction Grammar framework also facilitates its use for mapping previously unseen utterances to PropBank roleset instances. In order to evaluate the performance of our method on this semantic frame extraction task, we split the corpus into a test portion of 1,000 randomly selected utterances and a train portion of the remaining 153,391 utterances. We learn a new grammar from the training set, apply it to the utterances of the test set, and evaluate the extracted rolesets against the gold standard annotations of the corpus. Our method obtains a word-level precision of 76.15% and a recall of 76.36%, yielding an F_1-score of 76.25. When evaluating on the level of frames rather than rolesets, where it suffices to predict the correct frame rather than the exact roleset (e.g. `have` instead of `have.03` or `have.07`), it obtains a word-level precision of 79.85% and a recall of 80.07%, yielding an F_1-score of 79.96. Note that this evaluation establishes the baseline performance of a 'pure' grammar network, without any task-specific optimisation such as fine-tuning of heuristics or elimination of low-frequency and greedy constructions, which falls outside the scope of this paper.

4 Integration with PyFCG

The method presented in this paper has been integrated into the open-source Fluid Construction Grammar framework through the `fcg-propbank` subsystem. We briefly demonstrate its usage through the PyFCG Python library. A more extensive walk-through tutorial is available at `https://fcg-net.org/fcg-propbank`.

Let us first consider the use of an off-the-shelf, pre-trained grammar. After importing and initialising the PyFCG module, we create an agent of class `fcg.PropBankAgent`. We download a pre-

[5] We compute the similarity of constructions in terms of their weighted cosine similarity, where two nodes with the exact same weighted links to all other nodes would be perfectly similar.

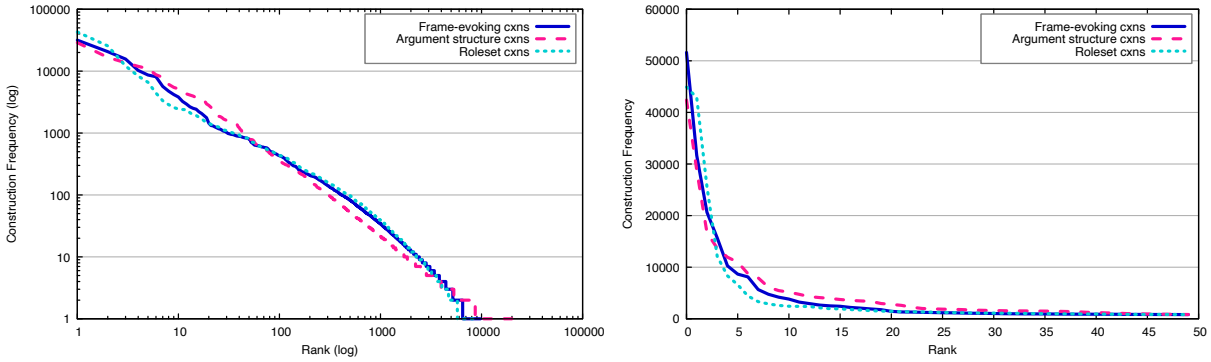

Figure 6: Frequency of constructions as a function of their rank in the frequency table, revealing their Zipfian distribution. Complete grammar network on log-log scale (left) and 50 most frequent constructions on linear scale (right).

compiled grammar for English from the FCG distribution and load it into our agent.

```
>>> import pyfcg as fcg
>>> fcg.init()
>>> pb_pretrained = fcg.PropBankAgent()
>>> f = fcg.load_resource('pb-en.fcg')
>>> pb_pretrained.load_grammar_image(f)
>>> pb_pretrained
<Agent: (id: agent-1) ~ 21052 cxns>
```

Our agent can now use its pretrained grammar to comprehend new utterances, such as "*Margaret Thatcher was elected Prime Minister of Britain.*" (*News on the Web* corpus 19-12-08-US; Herbst and Hoffmann, 2024). The resulting meaning representation reveals that the agent identified a single semantic frame that instantiates the `elect.01` PropBank roleset ('*elect someone to an office or position*'). The agent also understood that the roles of '*candidate*' (`arg1`) and '*office or position*' (`arg2`) in this instance of `elect.01` are respectively taken up by "*Margaret Thatcher*" and "*Prime Minister of Britain*".

```
>>> pb_pretrained.comprehend("Margaret
    Thatcher was elected Prime Minister
    of Britain.")
[{'roleset': 'elect.01', 'roles':
  [('v',"elected"), ('arg1',"Margaret
    Thatcher"), ('arg2',"Prime Minister
    of Britain")]}]
```

Let us now create a second agent, again as an instance of the `fcg.PropBankAgent` class, but let it learn a new grammar from corpus data instead of loading a pre-trained one. After having downloaded an example CoNNL file, in which a number of English sentences are annotated with PropBank rolesets[6], we call the agent's `learn_grammar_from_conll_file` method. This call initiates the learning process as described in Section 2 and equips the agent with the resulting grammar. In this case, the agent has learnt two frame-evoking constructions (for verbs with the lemmas *give* and *send*), two roleset constructions (for the rolesets `give.01` and `send.01`), and two argument structure constructions (a double object and a prepositional dative construction).

```
>>> pb_learner = fcg.PropBankAgent()
>>> f = fcg.load_resource('pb-
    annotations.conll')
>>> pb_learner.
    learn_grammar_from_conll_file(f)
>>> pb_learner
<Agent: (id: agent-2) ~ 6 cxns>
>>> list(pb_learner.grammar.cxns.keys())
['give(v)-cxn', 'send(v)-cxn',
 'give.01-cxn', 'send.01-cxn',
 'arg0(np)+v(v)+arg2(np)+arg1(np)-cxn',
 'arg0(np)+v(v)+arg1(np)+arg2(pp)-cxn']
```

We now instruct our agent to comprehend a previously unseen utterance, using the grammar it just learnt, by calling its `comprehend` method. While comprehending the utterance "*The King of the Belgians sent a box of chocolates to Forrest Gump.*", the agent identifies an instance of the `send.01` ('*give*') roleset, with "*The King of the Belgians*" as the '*sender*' (`arg0`), "*a box of chocolates*" as the '*thing sent*' (`arg1`) and "*to Forrest Gump*" as the '*sent-to*' entity (`arg2`).

```
>>> pb_learner.comprehend("The King of
    the Belgians sent a box of
    chocolates to Forrest Gump.")
[{'roleset': 'send.01', 'roles': [('v',
    "sent"), ('arg0', "The King of the
    Belgians"), ('arg1', "a box of
    chocolates"), ('arg2', "to Forrest
    Gump")]}]
```

[6]Due to licensing restrictions, we are not able to provide large PropBank-annotated corpora as downloadable PyFCG resources. We invite interested readers to obtain such corpora (e.g. OntoNotes or EWT) directly from the *Linguistic Data Consortium*.

5 Background and Related Work

Construction grammar has often been associated with the in-depth analysis of isolated linguistic phenomena that are in some way 'quirky' or at least difficult to account for in terms traditional grammatical analysis (cf. Fillmore et al., 1988; Jackendoff, 1997; Kay and Fillmore, 1999; Bergs, 2018; Hilpert and Bourgeois, 2020). However, as Fillmore (1988, p. 36) explains, the analysis of such "*non-central constructions*" in isolation was always envisioned as a means to achieve a deeper understanding of the "*mechanics of the grammar as a whole*" rather than constituting an end in itself. Today, efforts on scaling construction grammar are ongoing on three main fronts.

A first front is concerned with *constructicography* (Lyngfelt et al., 2018b), i.e. the curated inventorisation of the constructions of a language, along with the formulation of construction definitions and the annotation of example constructs. Constructicographical resources are currently being developed for different languages, including English (Fillmore et al., 2012), German (Ziem et al., 2019), Swedish (Lyngfelt et al., 2018a), Russian (Janda et al., 2020), Japanese (Ohara, 2018) and Brazilian Portuguese (Laviola et al., 2017). A second front is concerned with the formalisation and algorithmic processing of construction grammars. Beuls et al. (2021) present a human-engineered construction grammar for the extraction of causation frames from English texts. Micelli et al. (2009) and Dodge et al. (2017) leverage FrameNet to extend the coverage of human-engineered seed grammars, and van Trijp (2017) has developed a model that combines hand-crafted grammatical constructions with lexical items sourced from lexical resources. Approaches that have taken a more explicit usage-based perspective, by learning construction grammars from semantically annotated data (Chang, 2008; Gerasymova and Spranger, 2010; Spranger and Steels, 2015; Doumen et al., 2024; Beekhuizen, 2015) or from situated communicative interactions (Nevens et al., 2022; Beuls and Van Eecke, 2024), have so far only yielded grammars with a very limited coverage (see Doumen et al., 2025). A third front is concerned with the automatic extraction of re-occurring patterns from corpora of language use. Originally, data-driven methods were developed that learn partially abstract, re-occurring sequences of words and part-of-speech tags (Wible and Tsao, 2010; Forsberg et al., 2014; Barteld and Ziem, 2020). More recent methods extract minimal sets of constructional patterns that can combine information across different levels of linguistic analysis as annotated in a corpus or obtained through distributional methods (Dunn, 2017, 2022).

In addition, a diverse body of relevant work is somewhat more distantly related. The method of *collostructional analysis* (Stefanowitsch and Gries, 2003) statistically models the attraction and repulsion of lexemes with respect to grammatical patterns of interest as observed in corpus data. The *CASA* framework (Herbst and Hoffmann, 2024) provides a pedagogical method for systematically analysing utterances in terms of the constructions they instantiate. Popular annotation schemata have been extended to better capture information at the constructional level, in particular Abstract Meaning Representation (Bonial et al., 2018, 2026) and Universal Dependencies (Weissweiler et al., 2024). Finally, it has in the meantime become commonplace to investigate the linguistic capabilities of large language models using construction grammar as an underlying framework (Tayyar Madabushi et al., 2020; Weissweiler et al., 2022; Bonial and Tayyar Madabushi, 2024; Boguraev et al., 2025).

6 Conclusion

This paper has introduced a method for learning large-scale, broad-coverage construction grammars from corpora of language use. Starting from utterances annotated with constituency structure and semantic frames, the method facilitates the learning of vast networks of human-interpretable constructions that capture the relationship between syntactic structures and the frame-semantic relations they express. We have applied the method to a multi-genre corpus of English texts, have shown how the learnt grammars can both qualitatively and quantitatively support the usage-based, constructionist study of language, and have demonstrated the grammar's operationality by using it for semantic frame extraction on new data.

We are convinced that the trove of information embedded in these grammars offers a new range of opportunities for large-scale, usage-based construction grammar research. To support this, we have released with this paper a series of pre-trained grammars, accompanied by scripts for processing new data using existing grammars, for learning new grammars from semantically annotated corpora, and for linguistically analysing learnt grammars.

Limitations

The main limitation of the method introduced in this paper concerns its reliance on an external constituency parser, which at times can introduce inconsistencies that are difficult to recover from downstream. In particular, the method cannot correctly identify PropBank role instances in utterances wherever these do not correspond to constituents in the trees yielded by the parser.

We have so far only applied the method to English corpora that were annotated with constituency structures and PropBank rolesets. Extensions of the method to other widespread syntactic and semantic annotation schemata, such as Universal Dependencies and Abstract Meaning Representation, remain to be explored. Such extensions would be particularly useful when addressing languages that are typically not analysed in terms of constituency structure, but for which other utterance-level syntactic and semantic annotations are available.

The PropBank annotation scheme always associates rolesets with specific lexical items. Aspects of meaning that are contributed by non-substantive constructions, e.g. in the case of resultative ("*Firefighters cut the child free.*"), depictive ("*He left the bar drunk.*") or caused-motion constructions ("*The teacher shouted the children into a queue.*") remain unannotated in the training corpus and are consequently not picked up by the grammar. Ongoing work on the integration of constructional rolesets into AMR (cf. Bonial et al., 2018, 2026) shows promise in overcoming this limitation.

The evaluation of the method on the task of semantic role extraction that we reported on in the paper was done solely with the goal of demonstrating the coverage of a 'pure' grammar on new data. The results should be considered a baseline that was obtained without task-specific optimisations or fine-tuning of particular heuristics (cf. Van Eecke et al., 2022).

Acknowledgments

We are grateful to Frederik Cornillie, Anaïs Tack, Luc Steels, Remi van Trijp, Arno Temmerman and Jamie Wright for their insightful comments on an earlier version of this manuscript. This research was supported by the F.R.S.-FNRS-FWO WEAVE project HERMES I under grant numbers T002724F (F.R.S.-FNRS) and G0AGU24N (FWO), the Flemish Government under the Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen programme and the AI Flagship project ARIAC by DigitalWallonia4.ai.

References

Afra Alishahi and Suzanne Stevenson. 2008. A computational model of early argument structure acquisition. *Cognitive Science*, 32(5):789–834.

Collin Baker, Charles Fillmore, and John Lowe. 1998. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics.

Fabian Barteld and Alexander Ziem. 2020. Construction mining: Identifying construction candidates for the german constructicon. *Belgian Journal of Linguistics*, 34(1):5–16.

Barend Beekhuizen. 2015. *Constructions Emerging*. Ph.D. thesis, Universiteit Leiden.

Giulia Bencini and Adele E. Goldberg. 2000. The contribution of argument structure constructions to sentence meaning. *Journal of Memory and Language*, 43(4):640–651.

Alexander Bergs. 2018. Because science! notes on a variable. *Subordination in English: Synchronic and diachronic perspectives*, 101:43.

Katrien Beuls and Paul Van Eecke. 2023. Fluid Construction Grammar: State of the art and future outlook. In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, pages 41–50. Association for Computational Linguistics.

Katrien Beuls and Paul Van Eecke. 2024. Humans learn language from situated communicative interactions. What about machines? *Computational Linguistics*, 50(4):1277–1311.

Katrien Beuls, Paul Van Eecke, and Vanja Sophie Cangalovic. 2021. A computational construction grammar approach to semantic frame extraction. *Linguistics Vanguard*, 7(1):20180015.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English web treebank ldc2012t13. Philadelphia, Linguistic Data Consortium.

Hans C. Boas. 2025. Constructional syntax. In Mirjam Fried and Kiki Nikiforidou, editors, *The Cambridge Handbook of Construction Grammar*, pages 44–70. Cambridge University Press, Cambridge, United Kingdom.

Rens Bod. 2009. Constructions at work or at rest? *Cognitive Linguistics*, 20(1):129–134.

Sasha Boguraev, Christopher Potts, and Kyle Mahowald. 2025. Causal interventions reveal shared structure

across English filler–gap constructions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25032–25053.

Claire Bonial, Bianca Badarau, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Tim O'Gorman, Martha Palmer, and Nathan Schneider. 2018. Abstract meaning representation of constructions: The more we include, the better the representation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association.

Claire Bonial, Claire Benet Post, Paul Van Eecke, Katrien Beuls, and Harish Tayyar Madabushi. 2026. CxGr-AMR: Extending abstract meaning representation beyond lexically anchored relations with constructional rolesets. In *Proceedings of The Seventh International Workshop on Designing Meaning Representations (DMR 2026) @ LREC 2026*, pages 1–19.

Claire Bonial and Harish Tayyar Madabushi. 2024. Constructing understanding: on the constructional information encoded in large language models. *Language Resources and Evaluation*.

Nancy Chang. 2008. *Constructing grammar: A computational model of the emergence of early constructions*. Ph.D. thesis, University of California, Berkeley, Berkeley, CA, USA.

William Croft. 2001. *Radical construction grammar: Syntactic theory in typological perspective*. Oxford University Press, Oxford, United Kingdom.

Holger Diessel. 2015. Usage-based construction grammar. In Ewa Dabrowska and Dagmar Divjak, editors, *Handbook of Cognitive Linguistics*, pages 295–321. De Gruyter Mouton, Berlin, Germany.

Ellen Dodge, Sean Trott, Luca Gilardi, and Elise Stickles. 2017. Grammar scaling: Leveraging FrameNet data to increase embodied construction grammar coverage. In *The 2017 AAAI Spring Symposium Series*, pages 154–162, Washington, DC, USA. AAAI Press.

Jonas Doumen, Katrien Beuls, and Paul Van Eecke. 2024. Modelling constructivist language acquisition through syntactico-semantic pattern finding. *Royal Society Open Science*, 11(7):231998.

Jonas Doumen, Veronica J. Schmalz, Katrien Beuls, and Paul Van Eecke. 2025. The computational learning of construction grammars: State of the art and prospective roadmap. *Constructions and Frames*, 17(1):141–174.

Jonathan Dunn. 2017. Computational learning of construction grammars. *Language and Cognition*, 9(2):254–292.

Jonathan Dunn. 2022. Exposure and emergence in usage-based grammar: Computational experiments in 35 languages. *Cognitive Linguistics*, 33(4):659–699.

Charles J. Fillmore. 1976. Frame semantics and the nature of language. In *Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech*, volume 280 (1), pages 20–32.

Charles J. Fillmore. 1988. The mechanisms of "construction grammar". In *Proceedings of the Fourteenth Annual Meeting of the Berkeley Linguistics Society*, volume 14, pages 35–55. Linguistic Society of America.

Charles J. Fillmore, Paul Kay, and Mary Catherine O'Connor. 1988. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, 64(3):501–538.

Charles J. Fillmore, Russell R. Lee-Goldman, and Russell Rhodes. 2012. The FrameNet constructicon. In Hans C. Boas and Ivan A. Sag, editors, *Sign-based construction grammar*, pages 309–372. CSLI Publications/Center for the Study of Language and Information, Stanford, CA, USA.

Markus Forsberg, Richard Johansson, Linnéa Bäckström, Lars Borin, Benjamin Lyngfelt, Joel Olofsson, and Julia Prentice. 2014. From construction candidates to constructicon entries: An experiment using semi-automatic methods for identifying constructions in corpora. *Constructions and Frames*, 6(1):114–135.

Mirjam Fried. 2015. Construction grammar. In Tibor Kiss and Artemis Alexiadou, editors, *Syntax – Theory and Analysis: An International Handbook*, pages 974–1003. De Gruyter Mouton, Berlin, Germany.

Mirjam Fried and Kiki Nikiforidou, editors. 2025. *The Cambridge Handbook of Construction Grammar*. Cambridge University Press, Cambridge, United Kingdom.

Mirjam Fried and Jan-Ola Östman. 2004. Construction grammar: A thumbnail sketch. In Jan-Ola Östman and Mirjam Fried, editors, *Construction grammar in a cross-language perspective*, pages 1–86. John Benjamins, Amsterdam, Netherlands.

Kateryna Gerasymova and Michael Spranger. 2010. Acquisition of grammar in autonomous artificial systems. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI-2010)*, pages 923–928, Amsterdam, Netherlands. IOS Press.

Adele E. Goldberg. 1995. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press, Chicago, IL, USA.

Adele E. Goldberg. 2003. Constructions: A new theoretical approach to language. *Trends in Cognitive Sciences*, 7(5):219–224.

Adele E. Goldberg. 2019. *Explain me this*. Princeton University Press, Princeton, NJ, USA.

Adele E. Goldberg and Ray Jackendoff. 2004. The English resultative as a family of constructions. *Language*, 80(3):532–568.

Thomas Herbst and Thomas Hoffmann. 2024. *A Construction Grammar of the English Language: CASA - a Constructionist Approach to Syntactic Analysis*. John Benjamins Publishing Company, Amsterdam/Philadelphia.

Martin Hilpert. 2014. *Construction grammar and its application to English*. Edinburgh University Press, Edinburgh, United Kingdom.

Martin Hilpert and Samuel Bourgeois. 2020. Intersubjectification in constructional change: From confrontation to solidarity in the sarcastic much? construction. *Constructions and Frames*, 12(1):96–120.

Thomas Hoffmann. 2022. *Construction Grammar*. Cambridge University Press, Cambridge, United Kingdom.

Ray Jackendoff. 1997. Twistin' the night away. *Language*, 73(3):534–559.

Laura A. Janda, Anna Endresen, Valentina Zhukova, Daria Mordashova, and Ekaterina Rakhilina. 2020. How to build a constructicon in five years: the Russian example. *Belgian Journal of Linguistics*, 34(1):161–173.

Paul Kay and Charles Fillmore. 1999. Grammatical constructions and linguistic generalizations: The *What's X Doing Y?* construction. *Language*, 75(1):1–33.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2676–2686.

Andras Kornai. 2008. *Mathematical Linguistics*. Springer, London, United Kingdom.

Ronald Langacker. 1987. *Foundations of cognitive grammar: Theoretical prerequisites*, volume 1. Stanford University Press, Stanford, CA, USA.

Ronald Langacker. 2003. Constructions in cognitive grammar. *English Linguistics*, 20(1):41–83.

Ronald Langacker. 2008. *Cognitive Grammar: A Basic Introduction*. Oxford University Press, New York, NY, USA.

Adrieli Laviola, Ludmila Lage, Natalia Marcao, Tatiane Tavares, Vania Almeida, Ely Matos, and Tiago Torrent. 2017. The brazilian portuguese constructicon: Modeling constructional inheritance, frame evocation and constraints in framenet brasil. In *The 2017 AAAI Spring Symposium Series*, pages 193–196, Washington, DC, USA. AAAI Press.

Benjamin Lyngfelt, Linnea Backstrom, Lars Borin, Anna Ehrlemark, and Rudolf Rydstedt. 2018a. Constructicography at work: Theory meets practice in the swedish constructicon. In Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago T. Torrent, editors, *Constructicography: Constructicon development across languages*, pages 41–106. John Benjamins, Amsterdam, Netherlands.

Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago T. Torrent, editors. 2018b. *Constructicography: Constructicon Development across Languages*. John Benjamins, Amsterdam, Netherlands.

Kirsten Malmkjaer, editor. 2004. *The Linguistics Encyclopedia*. Routledge, London/New York. Second Edition.

Vanessa Micelli, Joachim De Beule, and Remi van Trijp. 2009. Framing Fluid Construction Grammar. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 31, pages 3023–3027. Cognitive Science Society.

Laura A. Michaelis. 2012. Making the case for construction grammar. In Hans C. Boas and Ivan A. Sag, editors, *Sign-Based Construction Grammar*, pages 31–69. CSLI Publications, Stanford, CA, USA.

Marvin Minsky. 1974. A framework for representing knowledge. Technical report, Technical Report 306, MIT AI Laboratory, Cambridge, MA, USA.

Jens Nevens, Jonas Doumen, Paul Van Eecke, and Katrien Beuls. 2022. Language acquisition through intention reading and pattern finding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 15–25. International Committee on Computational Linguistics.

Kyoko Ohara. 2018. Relations between frames and constructions: A proposal from the japanese framenet constructicon. In Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago T. Torrent, editors, *Constructicography: Constructicon development across languages*, pages 141–164. John Benjamins, Amsterdam, Netherlands.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Steven T. Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130.

Sameer Pradhan, Julia Bonn, Skatje Myers, Kathryn Conger, Tim O'gorman, James Gung, Kristin Wright-bettner, and Martha Palmer. 2022. PropBank comes of Age—Larger, smarter, and more diverse. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 278–288.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Michael Spranger and Luc Steels. 2015. Co-acquisition of syntax and semantics: an investigation in spatial language. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 1909–1915, Washington, DC, USA. AAAI Press.

Luc Steels. 2004. Constructivist development of grounded construction grammar. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 9–16. ACL.

Anatol Stefanowitsch and Stefan Th Gries. 2003. Collostructions: Investigating the interaction of words and constructions. *International Journal of Corpus Linguistics*, 8(2):209–243.

Harish Tayyar Madabushi, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. CxGBERT: BERT meets construction grammar. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4020–4032. International Committee on Computational Linguistics.

Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, Harvard, MA, USA.

Paul Van Eecke and Katrien Beuls. 2025. PyFCG: Fluid Construction Grammar in Python. *arXiv preprint arXiv:2505.12920*.

Paul Van Eecke, Jens Nevens, and Katrien Beuls. 2022. Neural heuristics for scaling constructional language processing. *Journal of Language Modelling*, 10(2):287–314.

Remi van Trijp. 2017. A computational construction grammar for English. In *The 2017 AAAI Spring Symposium Series*, pages 266–273, Washington, DC, USA. AAAI Press.

Remi van Trijp, Katrien Beuls, and Paul Van Eecke. 2022. The FCG Editor: An innovative environment for engineering computational construction grammars. *PLOS ONE*, 17(6):e0269708.

Lara Verheyen, Jonas Doumen, Paul Van Eecke, and Katrien Beuls. 2025. You shall know a construction by the company it keeps: Computational construction grammar with embeddings. In *Proceedings of the Second International Workshop on Construction Grammars and NLP*, pages 75–83. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. Ontonotes release 5.0 ldc2013t19. Philadelphia, Linguistic Data Consortium.

Leonie Weissweiler, Nina Böbel, Kirian Guiller, Santiago Herrera, Wesley Scivetti, Arthur Lorenzi, Nurit Melnik, Archna Bhatia, Hinrich Schütze, Lori Levin, Amir Zeldes, Joakim Nivre, William Croft, and Nathan Schneider. 2024. UCxn: Typologically informed annotation of constructions atop Universal Dependencies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16919–16932.

Leonie Weissweiler, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. The better your syntax, the better your semantics? Probing pretrained language models for the English comparative correlative. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10859–10882. Association for Computational Linguistics.

David Wible and Nai-Lung Tsao. 2010. StringNet as a computational resource for discovering and investigating linguistic constructions. In *Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics*, pages 25–31.

Alexander Ziem, Johanna Flick, and Phillip Sandkühler. 2019. The German constructicon project: Framework, methodology, resources. *Lexicographica*, 35(2019):15–40.

George Kingsley Zipf. 1936. *The Psychobiology of Language*. Routledge, London, United Kingdom.

George Kingsley Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, New York, NY, USA.

A Grammar Report

```
-------------------------------------------
GRAMMAR REPORT FOR PROPBANK-LEARNED
(<hashed-fcg-construction-set: 40,688 cxns>)
-------------------------------------------
Number of constructions:
   All cxns: 40,688
   Frame-evoking cxns: 9800
   Argument structure cxns: 22,568
   Roleset cxns: 8320
-------------------------------------------
Individual construction frequency information:
 All cxns:
    Absolute frequency: 1,117,581
    Mean frequency: 27.47
    Median frequency: 2
    Number of non-hapax cxns:  21,004 of 40,688
 Frame-evoking cxns:
    Absolute frequency: 372,527
    Mean frequency: 38.01
    Median frequency: 3
    Number of non-hapax cxns:  6,523 of 9,800
 Argument structure cxns:
    Absolute frequency: 372,527
    Mean frequency: 16.51
    Median frequency: 1
    Number of non-hapax cxns: 8,703 of 22,568
 Roleset cxns:
    Absolute frequency: 372,527
    Mean frequency: 44.77
    Median frequency: 3
```

```
    Number of non-hapax cxns: 5,778 of 8,320
-------------------------------------------
Construction network information:
   Average degree (argst-roleset): 6.35
   Average degree (fe-roleset): 1.46
   Average degree (fe-argst): 5.47
-------------------------------------------
```