Constructions 16 (2024)
doi 10.24338/cons-651

Evaluating Large-Scale Construction Grammars on the Tasks of Semantic Frame Extraction and Semantic Role Labeling

Thomas Moerman[1], Paul Van Eecke[2*], Katrien Beuls[3*]

[1]Ghent University, Language and Translation Technology Team (LT3), Belgium
[2]Vrije Universiteit Brussel, Artificial Intelligence Laboratory, Belgium
[3]Université de Namur, Faculté d'Informatique, Belgium
[*]Joint last authors

Abstract

This study investigates the application of computational construction grammars to semantic frame extraction (SFE) and semantic role labeling (SRL) in natural language processing. It provides an evaluation framework for large-scale construction grammars, combining the assessment of different simulations of sentence comprehension in the context of an SRL task. The study evaluates these grammars, which employ different configurations of heuristics - shortcuts or "rules of thumb" that guide how the grammar processes and understands language. Just as humans use various strategies to comprehend sentences quickly, these computational heuristics help the grammar efficiently apply constructions to extract meaning from an utterance. The focus lies on testing linguistically motivated heuristics, including (1) a preference for constructions that occur more often in language use, (2) a bias towards relating nearby elements in a sentence, and (3) a strategy that considers the strength of associations between different constructions in the linguistic network. The analysis reveals that different heuristics perform better in processing different semantic frames, highlighting the importance of well-designed heuristics. Notably, the heuristic that leverages the interconnectedness of constructions demonstrated superior overall performance, particularly on frames with high support, such as those with communication, cognition, and perception verbs. This suggests that considering how linguistic elements relate to each other in a broader network is crucial for accurate processing, especially in commonly occurring frames. The heuristic favoring local relationships showed effectiveness in frames involving expressions of desire, necessity, intention, and actions requiring immediate relationships between agents and actions. In contrast, relying solely on how frequently a construction occurs showed subpar performance across most frames compared to the other examined methods. These findings, derived from the application of the evaluation framework, contribute to the further development and operationalization of large-scale computational construction grammars, particularly in optimizing heuristic selection based on semantic frame properties.

Corresponding author(s): Thomas Moerman (thomas.moerman@ugent.be)

1 Introduction

Since the mid-1970s, the concept of "frames" has served as a valuable tool for understanding how language represents situations and events. This idea originated in various fields, including sociology (Goffman 1974), artificial intelligence (AI) (Minsky 1974), and linguistics (Fillmore 1976). Within linguistics, Fillmore's frame semantics emerged as a key method for analyzing how language represents events and situations. This theory posits that specific words or phrases evoke mental images, defining the roles and interactions among participants in a scenario. Therefore, understanding the meaning of a word requires knowledge of the entire context it relates to (Fillmore 2006; Boas 2021). For instance, comprehending "sell" necessitates familiarity with the commercial transfer context involving semantic roles (i.e., frame elements) like seller, buyer, goods, money, and their interrelations. By structuring these evoked frames, and specifying the event and participant roles, frame semantics, offers insights into how language encodes situational knowledge.

Leveraging this understanding of frames, tasks like semantic frame extraction (SFE) and semantic role labeling (SRL) emerged in natural language processing (NLP) to automatically extract and comprehend meaning from text. While SFE pinpoints the underlying event or situation being described and the participants involved, SRL identifies the roles played by different constituents within a sentence (e.g., who did what to whom) (Màrquez et al. 2008; Jurafsky and Martin 2020, 2:373). These tasks are crucial for natural language understanding (NLU), a broader field dedicated to developing systems that can understand natural language (Surdeanu et al. 2007).

Among various approaches to these tasks, Construction Grammar (CxG) offers a promising theoretical background to tackle SRL and SFE. Unlike traditional machine learning approaches that rely primarily on distributional semantics (He et al. 2018b; Mickus et al. 2019), CxG views language as a collection of constructions or form-meaning pairings (Fillmore, Kay, and O'Connor 1988; Kay 1990; Goldberg 1995; Croft 2001). By identifying the constructions used in a sentence, it is possible to infer the frame and participant roles, offering a transparent and linguistically motivated approach (Beuls, Van Eecke, and Cangalovic 2021). This makes large-scale, broad-coverage computational construction grammars possibly advantageous, as they have the potential to provide a deeper understanding of textual data compared to models based on distributional semantics.

This study evaluates how effectively CxG can be applied to identify semantic roles and frames within English sentences. It does this by analyzing the performance of computational construction grammars on a predefined task, namely SFE. A "construction grammar" is to be understood as a structured collection of form-meaning pairs (=constructions) that can be used to comprehend or formulate language (van Trijp, Beuls, and Van Eecke 2022). The study utilizes Fluid Construction Grammar[1] (FCG) (Steels 2011, 2017; Beuls and Van Eecke 2023) for formalizing and implementing the constructions. FCG is a computational framework that not only enables the semantic parsing of linguistic utterances through constructions but also provides the capability to learn grammars from semantically annotated corpora. This learned

[1] https://www.fcg-net.org/

grammar can then be applied for semantic parsing tasks, such as SFE, demonstrating FCG's versatility in both grammar learning and application. Therefore, the main goals of this study are to conduct an in-depth internal assessment of this methodology, to exemplify how CxG can be operationalized on a large scale through learning from annotated data, and to provide a systematic framework for evaluating construction grammar performance on a predefined semantic parsing task. Crucially, it focuses on the role of heuristics in this process. In this context, heuristics are computational shortcuts or "rules of thumb" that guide how the grammar processes and understands language. The study examines linguistically motivated heuristics, including a preference for frequently occurring constructions based on usage-based theories of language (Bybee 2006), local dependencies within sentences drawing from psycholinguistic research on sentence processing (Gibson 2000), and strengths in network connection between constructions aligning with cognitive linguistic theories (Diessel 2019, 2023). By evaluating the effectiveness of different heuristics in computational construction grammars through this framework, this study aims to offer insights that could inform both linguistic theory and constructional approaches to language processing, particularly in the context of learning and applying grammars for semantic parsing tasks.

AI's interest in studying language is not a new development. It could be argued that at the most fundamental level, the ability to communicate through language is a critical component of human intellect and, thus, a prerequisite for achieving the goal of AI to gain a similar range of abilities as humans (Russell and Norvig 2021, 823–24). One of the main challenges for AI to reach the ability to understand natural language is that it cannot be accomplished without necessary contextual knowledge and reasoning (Eisenstein 2019, 3–4). Although it appears to be an entirely different research field, Cognitive Linguistics and its extension, Cognitive Construction Grammar (CogCxG) (Goldberg 1995, 2006; Croft and Cruse 2004; Geeraerts 2006), share a common understanding with AI. Both fields agree that to interpret or generate natural language expressions, knowledge about the world is necessary, as the meaning of an expression depends on the situation in which it is uttered. In recent years, there has been a resurgent interest in unifying these disciplines, particularly in the field of Computational Construction Grammar (CCxG) (Beuls and van Trijp 2016; Nevens, Van Eecke, and Beuls 2019; Van Eecke and Beuls 2018; van Trijp 2017; Bergen and Chang 2005; Barres 2017; Beuls and Van Eecke 2024). This study contributes to this alignment by evaluating the effectiveness of computational construction grammars, a bridge between the two fields, in extracting semantic roles and frames from English sentences. Each grammar is configured with various learning and comprehension settings are analyzed. Performance is assessed on multiple levels, from overall to frame-specific effectiveness.

The resources and code used in this study can be found in the study's repository[2]. It is meant to supplement the main body of text and serves mainly as a technical documentation of the evaluation process. It contains all the code used to evaluate the frame predictions of large-scale construction grammars. The source code of the grammars themselves is publicly available through the Babel toolkit[3].

2 https://github.com/TomMoeras/evaluating-large-scale-construction-grammar-2024
3 https://emergent-languages.org/

This introduction is followed by the "Background" section, which provides the necessary background to the SRL and SFE task and the CCxG approach to this task. The "Methodology" section details the learning and comprehension processes of various grammars and how they are evaluated. The "Results" section breaks down the grammars' performance in macro- and frame-level evaluations, providing a thorough understanding of their effectiveness. The "Discussion" section interprets these findings in the context of ongoing construction grammar development, setting the stage for future research avenues. Finally, the "Conclusion" section summarizes the study's main findings and their implications.

To ensure this study remains accessible, the explanation of how the computational construction grammars were implemented and learned will be provided in a general, conceptual manner. For those seeking further details on their internal structure and processes, additional information, and explanations have been included in the appendix (A).

2 Background

This study is anchored in the domains of Frame Semantics, Construction Grammar (CxG), Semantic Role Labeling (SRL), and Semantic Frame Extraction (SFE). The section begins by detailing Frame Semantics and its resources, setting the foundation for understanding the problem space. It then highlights the linguistic motivations behind the approach, focusing on CxG and its relationship with Frame Semantics. Following this, the tasks of SRL and SFE are explained, along with commonly used methods. Finally, the section outlines the chosen approach, encompassing Computational Construction Grammar (CCxG) and its implementation within the Fluid Construction Grammar (FCG) framework.

2.1 Frame Semantics

Frame Semantics emphasizes the bond between form and meaning through semantic frames (Fillmore 2006; Boas 2021). These frames organize knowledge about word meanings, where understanding one concept implies grasping its broader context. For instance, the verb "to cook" can evoke an entire cooking scenario, simplifying it into a semantic frame. Within this frame, parts of the scene, termed frame elements, reflect cognitive comprehension of the experience. In language, these elements translate to semantic roles, defining the functions of entities in the described situation.

However, a semantic frame's representation in language is not always straightforward. A word can evoke multiple frames, underscoring the context-dependent nature of Frame Semantics. van Trijp (2024) outlines four mapping processes for expressing and understanding semantic frames:

1. Associating a situation with its relevant semantic frame.
2. Aligning frame elements with semantic roles.
3. Mapping semantic roles to grammatical functions.
4. Expressing these functions in specific linguistic choices.

Two major resources for frame semantics are PropBank (Palmer, Gildea, and Kingsbury 2005) and FrameNet (Ruppenhofer et al. 2006). PropBank, built upon the syntactic annotations of the Penn Treebank, provides a layer of semantic annotation that details verbs' roles and their associated arguments in the text. Each PropBank verb has numbered arguments (e.g., Arg0, Arg1) with specific semantic roles. On the other hand, FrameNet provides a more extensive inventory of semantic frames, covering a wide range of situations and concepts beyond just verbal predicates.

2.2 Construction Grammar

CxG views language as a collection of constructions or form-meaning pairings. It challenges traditional linguistic boundaries, treating all linguistic elements uniformly. While CxG has multiple theoretical offshoots, this short explanation focuses solely on aspects pertinent to the current study. For a broader understanding of CxG and its variations, works by Hoffmann (2022), Ungerer and Hartmann (2023), and van Trijp (2024) offer a comprehensive overview. The basic tenets of CxG are as follows:

- **Constructions as Linguistic Knowledge**: The foremost principle is that all linguistic knowledge can be expressed as constructions, or form-meaning pairings (Croft 2001; Croft and Cruse 2004; Fillmore 1985; Fillmore, Kay, and O'Connor 1988; Goldberg 1995; Kay 1990). These constructions freely interact to facilitate language comprehension and production, barring any conflicts.
- **Lexicon-Grammar Continuum**: CxG blurs the traditional boundaries between "words" and "grammar rules", treating all linguistic elements uniformly (Fillmore, Kay, and O'Connor 1988).
- **Dynamic Nature**: A notable aspect of (usage-based) CxG is its dynamic nature. Constructions are not innately available but are instead formed during communicative interactions. As they find success or failure in communication, constructions become more or less entrenched, enabling the grammar to represent individual linguistic knowledge rather than an idealized language user (Van Eecke and Beuls 2018).
- **Structured Inventory**: Lastly, CxG views an inventory of constructions as a structured entity rather than an unorganized list. It conceptualizes the constructions of a language as a structured collection, which can be depicted in the form of a network (Hoffmann 2017; Diessel 2019, 2023).

Understanding the essence of a "construction" in CxG is crucial for applying this linguistic theory. The discussions by Haspelmath (2023) and van Trijp (2024) contribute to this understanding. In this study, drawing from these discussions, a construction is understood to be a conventional schema for creating expressions, with at least one open slot that expressions from the same form-class can fill. This definition aligns with the view of constructions as recurrent, partially fixed patterns within a language, where certain elements are set while others allow for variability. Thus, in this study, a construction is considered a structure or template that guides the formation of linguistic units and is not an expression in itself. By extension, a construction grammar is a structured collection of constructions used to comprehend or formulate language.

The synergy between CxG and Frame Semantics is evident. Constructions, as form-meaning pairs, often evoke specific frames. This means that the meaning of a construction can be understood in terms of the frame it activates. Conversely, frames can be seen as providing the semantic backdrop against which constructions play out. For instance, a construction like "X sells Y to Z for W" evokes the "Commercial Transaction" frame, where X is the seller, Y is the item being sold, Z is the buyer, and W is the price. The construction lays out a specific morpho-syntactic pattern for expressing this frame.

2.3 Semantic Role Labeling and Semantic Frame Extraction

The task at hand is the prediction of PropBank roles and frames from English sentences. This labeling of semantic roles decodes sentences to answer questions like "Who is doing what to whom, when, where, and how?" (Jurafsky and Martin 2020, 2:373–74).

SRL is the task of identifying the predicates in a sentence and labeling the semantic relationships between these predicates and their associated arguments. For example, in the sentence "Jane sold her old car for $5000", SRL would identify "sold" as the predicate, "Jane" as the seller (Arg0), "her old car" as the item sold (Arg1), and "for $5000" as the price (Arg2). SFE is rooted in SRL principles. It identifies a word or phrase's semantic frame and categorizes the roles of other sentence elements in relation to this frame. For instance, in the sentence "Jane sold her old car for $5000," the verb "sold" triggers the "Commercial Transaction" frame, indicating roles such as the seller ("Jane"), the item sold ("her old car"), and the price ("$5000"). Using this approach, SRL systems detect frame-evoking elements and align other sentence elements with the frame's designated roles. The aim goes beyond identifying individual actions, striving to understand the overarching event conveyed by the sentence. Resources like FrameNet and PropBank facilitate this extraction with their extensive frame and role inventories.

To illustrate the typical SRL process (Pradhan et al. 2005; He et al. 2018a; Li et al. 2019) with PropBank annotations, consider the sentence "Jane sold her old car for $5000":

1. **Predicate Identification**: Predicate: "sold"
2. **Predicate Sense Disambiguation**: Predicate Sense: sell.01, "Transferred ownership in exchange for money."
3. **Argument Identification**: "Jane", "her old car", "for $5000"
4. **Argument Classification**: Classified Arguments: "Jane" (Arg0: Seller/Agent), "her old car" (Arg1: Item Sold/Theme), "for $5000" (Arg2: Price)

While the organization and formalization of semantic frames, including FrameNet's frames and PropBank's rolesets, on a large scale have been carried out with success, the automated extraction of these frames and rolesets from text presents a significant challenge (Beuls, Van Eecke, and Cangalovic 2021).

2.4 Approaches to SRL and SFE

Various approaches have been developed for SRL and SFE tasks. These methods generally rely on large-scale data analysis and statistical patterns rather than on explicit linguistic structures or theories of language. Commonly used methods include:

1. Feature-based models: These models use hand-crafted features derived from linguistic analysis (e.g., part-of-speech tags, dependency parse trees) to train machine learning classifiers (Pradhan et al. 2005). While these features are based on linguistic knowledge, the models themselves do not explicitly represent or reason about linguistic structures. Instead, they treat these features as statistical indicators for classification tasks.
2. Neural network-based models: Recent approaches leverage deep learning techniques, particularly transformer-based models like BERT, to learn representations of words and sentences for SRL and SFE tasks (Shi and Lin 2019). These models rely on vast amounts of data to learn patterns and relationships without incorporating explicit linguistic knowledge or structures. They operate on the principle that meaning can be derived from the distributional properties of words in large corpora rather than from predefined linguistic constructs.
3. Joint models: These approaches perform SRL along with other NLP tasks, such as syntactic parsing, aiming to leverage the interactions between different levels of linguistic analysis (He et al. 2018a). While these models acknowledge the interplay between different linguistic levels, they typically do not model language as a structured inventory of form-meaning pairings. Instead, they treat these interactions as statistical dependencies that can be learned from data.

These approaches have shown considerable success in SRL and SFE tasks, achieving high performance on standard benchmarks. For instance, the AllenNLP model, based on the BERT architecture, has achieved a test F1 score of 86.49 on the OntoNotes 5.0 dataset for English PropBank SRL (Gardner et al. 2017). However, these methods have certain limitations. They typically require large amounts of annotated data for training, which can be costly and time-consuming to produce. Moreover, their reliance on statistical patterns means they may struggle with rare or novel linguistic constructions not well-represented in their training data. The internal workings of these models, mainly neural network-based ones, can also be difficult to interpret, making it challenging to understand why they make specific predictions or how they represent linguistic knowledge.

This study focuses on a different approach: Computational Construction Grammar (CCxG). Unlike the above methods, CCxG aims to model language processing more closely to how humans might understand and produce language, providing a more transparent and interpretable model of language processing.

2.5 The Approach: Computational Construction Grammar

The approach adopted for the task of predicting PropBank frames is grounded in the principles of CCxG, particularly through the utilization of the Fluid Construction Grammar (FCG) framework (Steels 2011, 2017; Beuls and Van Eecke 2023, 2024). CCxG is an ambitious attempt to translate the foundational tenets of linguistic Construction Grammar into computational models. These models are designed to systematically capture linguistic knowledge, thus enabling the efficient processing of natural language input in a way that aligns with theoretical linguistic principles. A standout example of CCxG is FCG (Steels 2017). FCG presents a structured methodology for embedding construction grammars within a computational environment. Its capabilities span both the comprehension and production of language based on constructional principles.

FCG forms the foundation for the construction grammars analyzed in this study. The central question FCG engages with is: how can constructions be utilized to comprehend or formulate language computationally? In this context, comprehension refers to translating a natural language expression into a representation of its meaning, whereas formulation or production denotes the reverse process, turning a semantic representation into a natural language utterance (Van Eecke, Nevens, and Beuls 2022).

For constructions to facilitate comprehension or formulation, FCG treats language processing akin to a problem-solving task, an approach drawn from the field of AI (Russell and Norvig 2021, 63–104). Consider the example of solving a puzzle. The task is straightforward, namely, finding a path that leads to a completed puzzle. The problem-solver begins with an initial state (e.g., a jumbled puzzle), performs actions or operations (e.g., moving and rotating pieces), and aims to reach a desired goal state (e.g., a completed puzzle).

In FCG's context, these problem-solving elements manifest as transient structures and constructions. Transient structures serve as the state representation (i.e., a jumbled puzzle). They are mutable, temporary configurations that evolve during language processing. Constructions, on the other hand, are akin to operators (i.e., actions that change the state of the jumbled puzzle). They are applied to the transient structure, effecting changes and driving the progression from the initial state towards the goal.

In FCG, language processing can thus be characterized as starting with an initial transient structure and sequentially applying a series of constructions to resolve a language comprehension or formulation problem. The role of the FCG engine is to determine a pathway from the initial transient structure to the final transient structure, a path that effectively leads to successful language comprehension or formulation. The system utilizes various strategies to identify this pathway, with heuristics being the most pertinent for this study. These heuristics guide the process, providing cues on which construction to apply next or which direction to take in the search tree exploration (Van Eecke 2018; Van Eecke, Nevens, and Beuls 2022). These strategies can be diverse and might be rooted in various linguistic indicators. For example, a heuristic might prioritize constructions that match more units in the transient structure or prefer constructions that are more frequently observed in the language data. Several heuristics will be individually assessed during the evaluation process to ascertain which one delivers the most effective results.

In computational applications, combining the insights from both CxG and Frame Semantics can provide a powerful tool for natural language understanding. Constructions offer a systematic way to represent and parse linguistic structures, while frames provide the semantic depth to comprehend and formulate meaningful content.

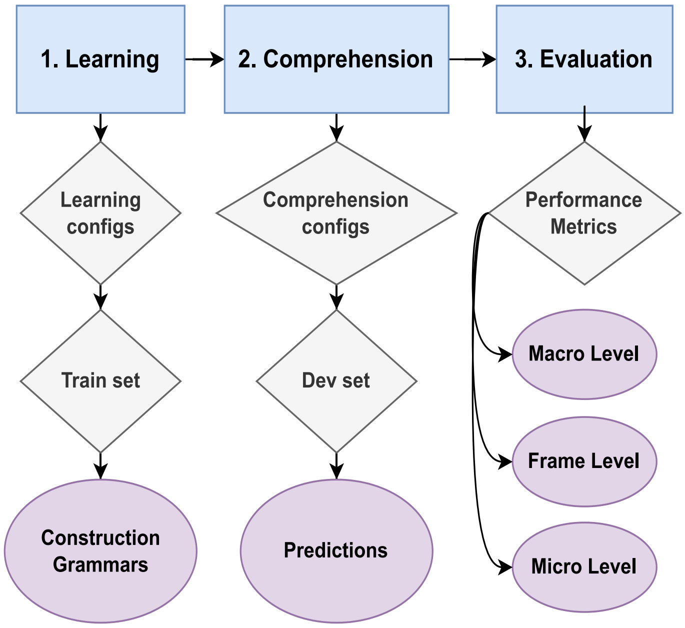

Figure 1: Evaluation Process of the Construction Grammars

3 Methodology

To analyze and evaluate the performance of the grammars, each with its unique learning and comprehension configurations, were examined. The evaluation process (Figure 1) comprises three main steps. Initially, grammars are learned by implementing a range of learning configurations. Subsequently, these grammars are utilized to comprehend and extract semantic frames from sentences, employing various heuristics. Lastly, the predictions obtained are assessed on different levels using multiple performance metrics.

The subsequent sections will delve deeper into each of these stages, with a predominant focus on the evaluation process, aligning with the study's main objectives.

3.1 Learning and Comprehension Process

The foundation for the large-scale construction grammars examined in this study lies in their learning process. This process involves acquiring knowledge about linguistic structures and developing an interconnected system that can subsequently be used for comprehension. The learning process for a PropBank construction grammar involves acquiring linguistic structures and developing a system for comprehension. In this study, PropBank annotations taken from the OntoNotes 5.0 corpus (Weischedel et al. 2013) are used as the primary data source, and the grammars undergo a structured learning process, extracting three primary types of constructions: Argument Structure Constructions, Word Sense Constructions, and Lexical Constructions. These constructions form the construction inventory. Constructions serve as mappings between form and meaning, with form represented by syntactic tree structures and meaning by semantic roles. Lexical Constructions identify specific verb forms; Word Sense Constructions specify the context or sense in which these verb forms are used; and Argument Structure Constructions map from a syntactic structure (form) to semantic roles (meaning) that are realized at specific places in the tree, establishing a direct link between the structure of a sentence and its semantic roles. Alongside the construction inventory, the learning process also develops a categorial network. This network represents the categories that are learned and their relationships. The constructions in the inventory make use of the categories in the categorial network during the comprehension process. Following comprehension, frames are extracted to provide a structured meaning representation of the depicted situations or events.

To maintain the focus of this discussion on the evaluation, the precise methodologies employed in the learning and operationalization of these grammars, as well as specific examples of these constructions are not elaborated upon here. However, for those interested in delving deeper into these processes, detailed explanations and insights are provided in the appendix. This additional information offers an overview of the underlying mechanisms that enable the grammars to effectively learn and apply linguistic knowledge.

The study employed a range of learning and comprehension settings, detailed in Table 1. Learning settings define what the grammar needs to learn, and comprehension settings guide the application of these learned constructions to get to a meaning representation. All explanations here are kept to a conceptual level to ensure readability, with additional details provided in the appendix.

This study aimed to investigate the effects of three key heuristics on grammar performance: (1) **Frequency-Based Priority**, (2) **Network Connection Strength**, and (3) **Local Dependency Preference** (collectively referred to as the H1 group). These heuristics were selected based on their potential to capture different aspects of language processing and construction grammar theory.

- The **Frequency-Based Priority** heuristic is grounded in usage-based theories of language acquisition and processing (Bybee 2006). It reflects the hypothesis that more frequently encountered linguistic patterns play a crucial role in language comprehension and production.
- The **Network Connection Strength** heuristic aligns with the conceptualization of constructions as part of a structured network, as proposed in cognitive linguistic theories (Hoffmann 2017; Diessel 2019, 2023). This heuristic reflects the interconnected nature of constructions in the grammar and the importance of these connections in language processing.
- The **Local Dependency Preference** heuristic draws from psycholinguistic research on sentence processing, particularly the idea of locality effects in language comprehension (Gibson 2000). It was chosen to test the importance of local dependencies in sentence comprehension within the framework of construction grammar.

The study implemented various learning configurations to examine the impact of these H1 heuristics. These configurations served two primary purposes:

1. By altering what the grammar needs to capture, these configurations fundamentally change the internal structure of the construction inventory and its associated network of relationships. This allows for an investigation into how such

Table 1: This table provides an overview of the learning and comprehension configurations used in the study. The 'Group' column indicates the group to which each configuration setting belongs. 'L' denotes Learning Configurations, and 'H' denotes Heuristics or comprehension configurations. The number following 'L' or 'H' represents the group number. '0' indicates that the setting is always constant. Numbers '1' and '2' denote mutually exclusive groups. For instance, in the Learning Config group 'L1', if one setting is activated, the other is turned off.

Learning Configurations		
Configuration	**Group**	**Description**
Core Roles	L0	Includes Arg0-1-2-3-4-5. Always included.
Modifier Group Inclusion	L1	Includes modifiers as defined by Palmer et al. (2005).
Modifier Group Exclusion	L1	Excludes modifiers from the learning process.
Full Roleset Inclusion	L2	No rolesets are excluded.
Frequent Multi-Sense Roleset Exclusion	L2	Excludes certain rolesets from the grammar's learning process, such as different word senses of the verbs "be", "have" and "get".
Heuristics		
Heuristic	**Group**	**Description**
Frequency-Based Priority	H1	Prioritizes frequently encountered constructions. This heuristic is based on the assumption that the constructions encountered more often in the data are more likely to be accurate or relevant.
Network Connection Strength	H1	Focuses on the strength of connections between constructions.
Local Dependency Preference	H1	Prioritizes connections between elements closer together in the sentence.

structural changes affect the grammar's overall performance and the efficacy of the H1 heuristics.

2. The inclusion or exclusion of specific elements enables an assessment of how well the heuristics perform in comprehending these particular components within sentences. This approach facilitates a direct comparison between grammars that include or exclude certain elements, providing insights into the heuristics' effectiveness across different linguistic contexts.

Two primary learning modes were employed: **Core Roles** and **Modifier Group**. The **Core Roles** mode focused on basic argument roles (Arg0-1-2-3-4-5). In contrast, the **Modifier Group** included additional modifiers such as temporal or locational information as defined in Palmer, Gildea, and Kingsbury (2005). Furthermore, the study introduced roleset categories: **Frequent Multi-Sense Roleset Exclusion** and **Full Roleset Inclusion**. The **Frequent Multi-Sense Roleset Exclusion** learning configuration primarily focuses on auxiliary verbs and frequently occurring verbs with multiple senses. This includes different word senses of the verbs 'be', 'have', 'do', and 'get'. These verbs were chosen for exclusion for several reasons. They appear in a wide variety of contexts and constructions. Each of these verbs has numerous senses or uses, ranging from auxiliary functions to main verb uses with distinct meanings. Due to their frequency and multiple senses, these verbs create a large number of connections within the network. This can significantly increase the complexity of the network and potentially obscure patterns related to less frequent but semantically rich verbs. By excluding these frequent multi-sense rolesets, the study aims to simplify the learning task and potentially improve performance on core verbal predicates.

Conversely, **Full Roleset Inclusion** provided a more comprehensive learning approach, incorporating all rolesets, including these frequent multi-sense verbs. This setting allows for an examination of how including these complex, highly connected verbs affects the overall performance of the grammar and the efficacy of different heuristics. The comparison between these two settings enables an assessment of the trade-offs between comprehensive coverage and focused learning in the context of construction grammar. It provides insights into how the presence or absence of these frequent multi-sense verbs impacts the grammar's ability to capture and utilize constructional patterns in language processing.

By structuring the study in this way, the aim was to evaluate the individual performance of each heuristic and learning configuration and uncover potential synergies or conflicts between them. This approach allows insights into how different aspects of language understanding, from fundamental argument structures to more nuanced linguistic features, interact within the framework of construction grammar. Moreover, it provides a comprehensive view of how changes in the grammar's internal structure and the inclusion or exclusion of specific linguistic elements affect the heuristics' performance and the grammar's overall effectiveness.

The grammars were learned using the OntoNotes 5.0 corpus train set (Weischedel et al. 2013). This corpus can be described as a broad-coverage corpus that spans several genres, including religious texts, telephone conversations, news articles and weblogs. The train set, used to learn the grammars, consisted of 123,648 sentences. Each of these sentences is annotated with a PropBank layer. The grammars acquired between 38,897 and 82,045 constructions through this learning process, the exact number varying based on the specific learning configurations employed. To evaluate the grammars' performance, a subset of 4,000 sentences (Subdev) was used from the development set (Dev) of the same corpus (Table 2). This approach allows for iterative grammar refinement based on performance and reduces the risk of overfitting, keeping the test set (Test) separate for final evaluation.

Random and shuffled sentences were used for evaluation, but future research could benefit from a more selective approach to curating the sentence selection to focus on specific aspects of the grammars' performance.

3.2 Evaluation Process

This section provides an overview of the evaluation criteria applied in assessing the performance of the grammars and the learning, and comprehension configurations. These criteria serve as the foundation for comparing the effectiveness of different grammars and configurations in the context of semantic frame extraction.

Table 2: Overview of sentences and frames in the corpus.

Dataset	#Sentences (S)	With Frames	Without Frames	% Without	Avg #Frames/S	Mode Frames
Train	123,648	107,585	16,063	12.99	3.30	2
Dev	17,058	14,808	2,250	13.19	3.28	2
Subdev	4,000	3,461	539	13.48	3.01	2
Test	13,685	11,531	2,154	15.74	3.22	2
Total	154,391	133,924	20,467	13.26	3.29	2

3.2.1 Metrics

The primary metric used in the evaluation process was the F1 score, which leverages both precision and recall[4] in its computation. The calculation of the F1 score was done using a function that compared predictions to a gold standard. This function was adapted to account for specific grammar configurations, ensuring consistency in the evaluation process:

- Word sense disambiguation was always included, ensuring that the evaluation considered the specific word senses in the context of the sentences.
- If certain role sets were excluded during the learning process, they were also excluded from the evaluation process to maintain consistency between learning and evaluation.
- When the **Modifier Group** was not learned as part of the grammar configuration, the predictions were evaluated solely on the **core roles**, allowing for a more focused assessment of grammar performance concerning these essential elements.

To illustrate this further, Table 3 presents a comparison between the gold standard frames and the frames predicted by various grammars for the sentence “Cathryn Rice could hardly believe her eyes”, focusing on the inclusion/exclusion of the **Modifier Group** configuration.

- **Gold Standard Frames**: Serve as the reference point for evaluating the accuracy of the grammars’ predictions. These include the primary frame “believe.01”, with “Cathryn Rice” as the agent, “could” as the modal, “hardly” as the adverb, “believe” as the verb (Frame-Evoking Element, FEE), and “her eyes” as the theme.
- **Grammar Predictions**:
  - **Grammar 1 (all roles)**: This grammar scores an F1 of 1, indicating a perfect match with the gold standard for all roles. This showcases its effective learning and prediction ability across the full range of roles.

[4] For the sentence “Cathryn Rice could hardly believe her eyes”, precision (P) measures the accuracy of the grammar’s predictions against the gold standard (e.g., identifying “believe” as the verb, “Cathryn Rice” as the agent, “her eyes” as the theme), calculated by dividing correct predictions by total predictions. Recall (R) assesses the grammar’s ability to identify all correct instances as per the gold standard, calculated by dividing the number of correct grammar predictions by the gold standard instances. If the grammar perfectly matches the gold standard’s expectations for all roles, both precision and recall, would be 100%.

Table 3: Comparison of gold standard and grammar predictions with F1 scores. Purple boxes signify the gold standard elements, serving as the target for prediction. Green boxes indicate correct predictions aligning with the gold standard. Grey boxes for Grammar 2 signify the absence of predictions for modifier roles, aligning with its learning constraints and not indicating an error. Red boxes in Grammar 3 signal inaccuracies in role prediction or classification.

	Gold Standard	Grammar 1 (all roles)	Grammar 2 (core roles)	Grammar 3 (all roles)
Frame	believe.01	believe.01	believe.01	believe.01
ARG0 (Agent)	Cathryn Rice	Cathryn Rice	Cathryn Rice	Cathryn Rice
MOD	could	could	$\emptyset$	$\emptyset$
ADV	hardly	hardly	$\emptyset$	$\emptyset$
V (FEE)	believe	believe	believe	believe
ARG1 (Theme)	her eyes	her eyes	her eyes	her eyes
MNR				could
EXT				hardly
F1 Score		1	1	0.71

- **Grammar 2 (core roles)**: Similarly scores an F1 of 1. The light grey boxes for *MOD* and *ADV* do not signal shortcomings but define its focus. Since Grammar 2 was not designed to learn modifier roles, its precision and recall are to be evaluated within the context of its specific learning parameters. Its performance is deemed perfect for this sentence, given its targeted learning scope.
- **Grammar 3 (all roles)**: Shows a divergence from the gold standard, especially in its treatment of modifier roles, resulting in a reduced F1 score of 0.71. This points to a struggle in fully capturing all the roles as intended, contrasting with the more accurate performances of Grammars 1 and 2.

By adapting the evaluation process according to the grammar configuration, the study ensured a fair and accurate assessment of each grammar's performance in extracting semantic frames.

3.2.2 Levels of Analysis

To comprehensively evaluate the performance of the construction grammars in extracting semantic frames, the study employs a multi-layered approach, encompassing macro and frame levels of analysis.

The macro evaluation provides an overall view of each grammar's performance across the entire corpus, moving beyond single sentence analysis to assess efficacy throughout the dataset. This broad perspective helps to identify the success of each grammar configuration, and to suggest optimal configurations for future uses.

At a more detailed level, the frame-level analysis looks at individual semantic frames. Using VerbAtlas (Di Fabio, Conia, and Navigli 2019), PropBank role sets are grouped based on shared semantic meanings to provide a clear and structured evaluation. VerbAtlas provides manually labeled mappings that group semantically similar PropBank role sets, adding depth to the assessment of grammar prediction capabilities. For example, the role sets "remain.01", "stay.01", and "keep.04" are grouped under the VerbAtlas frame "REMAIN", which then becomes the focal point for evaluating grammar performance related to these role sets. A weighted F1 score, adjusted for the support value of different role sets is used to ensure a balanced evaluation that avoids bias from variations in role set occurrences in the dataset.

3.2.3 Evaluating Configuration Settings: the Composite Performance Score (CPS)

The evaluation process centered on comparing performance metrics across various learning configurations and comprehension heuristics. This involved calculating differences between configuration settings' active and inactive states and deriving a composite score based on these differentials. Aggregation functions like mean, maximum, and minimum were pivotal in this evaluation. For instance, while the average value might give a general performance indicator, max and min values show the best and worst performances, respectively.

Table 4 showcases the grammars' performance for the frame "REMAIN" with its associated role sets. The table serves to exemplify the assessment method used throughout the study rather than presenting final results. This example illustrates how different learning configurations and comprehension heuristics affect the F1 score, both in their active (I) and inactive (O) states. The Composite Performance Score (CPS) is a synthesized score derived from these difference scores, adjusted by certain weights and multiplied by 10 for ease of interpretation. Weights were assigned to the mean (0.5), max (0.7), and min (0.3) aggregation functions to prioritize consistent performance, grammar potential, and worst-case scenarios, respectively.

When considering the "REMAIN" frame (Table 4), several key observations can be made across the learning configurations and heuristics. In the learning configurations, the L1 group (Modifier Group) shows a significant impact. **Modifier Group Exclusion** enhances performance with a CPS of +12.5, while **Modifier Group Inclusion** has an equally adverse effect with a CPS of -12.5. This suggests that excluding modifiers leads to better performance for the "REMAIN" frame. In the L2 group (Roleset Inclusion), the impact is less pronounced, with **Full Roleset Inclusion** slightly improving performance (CPS: +1.0) and **Multi-Sense Roleset Exclusion** slightly decreasing it (CPS: -1.0).

Among the heuristics, the H1 group shows significant variations. The **Local Dependency Preference** heuristic enhances sentence comprehension the most, with the highest CPS of +17.7. Grammars incorporating this heuristic outperformed others, showing a mean F1 score improvement of +0.15 and a minimum F1 score improvement of +0.34. The **Network Connection Strength** heuristic also indicates a positive impact (CPS: +7.1). However, the **Frequency-Based Priority** heuristic severely impairs performance with a CPS of -63.6, suggesting it should be disabled for optimal

Table 4: Composite Performance Scores (CPS) of learning configurations and comprehension heuristics for the VerbAtlas frame 'REMAIN' on the first level, (metric F1 score)

Configuration Settings	Δ Mean (I/O)	Δ Max (I/O)	Δ Min (I/O)	CPS
Learning Configurations				
L1: Modifier Group				
Modifier Group Exclusion	0.11	0.10	0.00	+12.5
Modifier Group Inclusion	-0.11	-0.10	0.00	-12.5
L2: Roleset Inclusion				
Full Roleset Inclusion	0.02	-0.00	-0.00	+1.0
Multi-Sense Roleset Exclusion	-0.02	0.00	0.00	-1.0
Heuristics				
H1				
Local Dependency Preference	0.15	-0.00	0.34	+17.7
Network Connection Strength	0.09	-0.07	0.25	+7.1
Frequency-Based Priority	-0.38	-0.53	-0.25	-63.6

results on the "REMAIN" frame.

The quantitative methodology employed here offered a multi-level evaluation of the grammars with varied configurations. Through macro- and frame-level evaluations, a holistic understanding of the grammars' strengths and areas for improvement emerged, setting the stage for further optimization of the large-scale PropBank construction grammars.

4 Results

This section breaks down the performance of construction grammars into two main areas: their overall effectiveness across the dataset and their accuracy in predicting specific semantic frames.

4.1 Macro-Level Evaluation

The performance of all grammars is evaluated based on three key metrics: precision, recall, and F1 score. Analysis of the performance metrics across all grammars reveals considerable variation in results. This variability indicates that the effectiveness of these grammars differs based on their configuration. For example:

- The highest-performing grammar, which excluded multi-sense rolesets and modifiers during learning and used **Network Connection Strength** as heuristic, achieved an F1 score of 0.70 (precision: 0.81, recall: 0.61).
- The best-performing grammar that learned all rolesets and modifier roles, us-

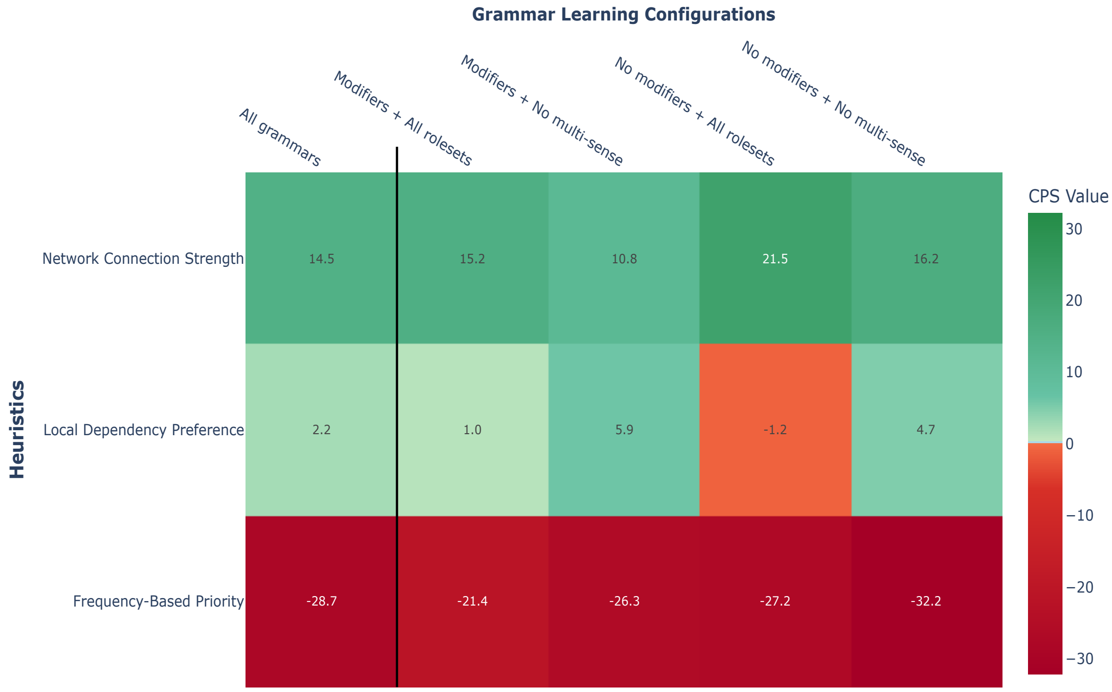

Figure 2: Heat-map of Heuristics and Grammar Configurations. The columns represent different learning configurations, from all grammars (leftmost) to specific combinations of modifier and frequent multi-sense roleset inclusion/exclusion. The rows show the performance of individual heuristics (H1 group). Color intensity indicates the magnitude of the CPS, with green representing positive impact (improvement in F1 score) and red representing negative impact (decrease in F1 score). Numbers in each cell represent the exact CPS value for that combination of heuristic and grammar configuration.

ing **Network Connection Strength** as heuristic, obtained an F1 score of 0.59 (precision: 0.66, recall: 0.53).

- In contrast, the lowest-scoring grammar, which learned all rolesets and modifier roles, employed **Frequency-Based Priority** as heuristic, achieved an F1 score of 0.38 (precision: 0.50, recall: 0.30).

It is essential to look into the impact of individual configuration settings on the performance of these grammars. By investigating the effects of specific settings, a better understanding of their contributions to the grammars' F1 scores can be gained. This can help to identify the most influential settings and possible interactions between them, ultimately guiding the development of more effective grammars.

The Composite Performance Score (CPS) results are presented in a heat-map (Figure 2), which provides insights into the effectiveness of these settings in terms of overall F1 score[5]. As described in Section 3.2.3, the CPS is a synthesized score that combines differences in F1 scores between active and inactive states of configuration settings, weighted to prioritize consistent performance, grammar potential, and worst-case scenarios.

In Figure 2, the columns represent different learning configurations, ranging from all grammars (leftmost column) to specific combinations of modifier and multi-sense

[5] Significance testing confirmed statistically significant differences ($p < 0.05$) between all pairs of key heuristics

roleset inclusion/exclusion (rightmost columns). The rows show the performance of individual heuristics from the H1 group. The color intensity in each cell indicates the magnitude of the CPS, with darker green representing a stronger positive impact (improvement in F1 score) and darker red representing a stronger negative impact (decrease in F1 score). The numbers in each cell represent the exact CPS value for that combination of heuristic and grammar configuration.

Looking at the figure, several key observations can be made:

1. **Network Connection Strength (H1):** This heuristic consistently shows the highest positive impact across all configurations, with CPS values ranging from 10.8 to 21.5. It performs particularly well in configurations without modifiers (two rightmost columns), suggesting it is most effective when dealing with core roles. The highest CPS value (21.5) is observed for the configuration without modifiers and including all rolesets.
2. **Local Dependency Preference (H1):** This heuristic shows variable performance across configurations. It has a notably positive impact (CPS: 5.9) when modifiers are included but frequent multi-sense rolesets are excluded (middle column). However, it shows a slight negative impact (-1.2) in configurations without modifiers and all rolesets included (second from right column), suggesting that it has difficulties dealing with these frequent multi-sense verbs when modifiers are absent.
3. **Frequency-Based Priority (H1):** This heuristic consistently shows the most substantial negative impact across all configurations, with CPS values ranging from -21.4 to -32.2. The negative impact is most pronounced (-32.2) in the configuration without modifiers and excluding multi-sense rolesets (rightmost column). This suggests that relying heavily on frequency information may lead to decreased precision and recall, regardless of the learning configuration, and this effect is exacerbated when dealing with a more restricted set of roles and rolesets.

These observations highlight the complex interactions between heuristics and grammar configurations. The **Network Connection Strength** heuristic appears to be the most robust across different configurations, while the effectiveness of **Local Dependency Preference** varies depending on the inclusion of modifiers and multi-sense rolesets. The consistent negative impact of the **Frequency-Based Priority** heuristic suggests that this approach may be less suitable for the task at hand, regardless of the grammar configuration.

4.2 Frame-Level Evaluation

The frame performance analysis is essential to evaluating the grammars' effectiveness in handling various PropBank frame rolesets. This analysis utilizes the VerbAtlas mappings to cluster PropBank frames into semantically related groups. By examining the performance of the grammars at the frame level, insights are gained into how well the grammars capture and generalize linguistic patterns within each frame.

The frame performance plot (Figure 5) displays the weighted mean F1 score against the coefficient of variation (CV) for each frame, with the size of the points representing

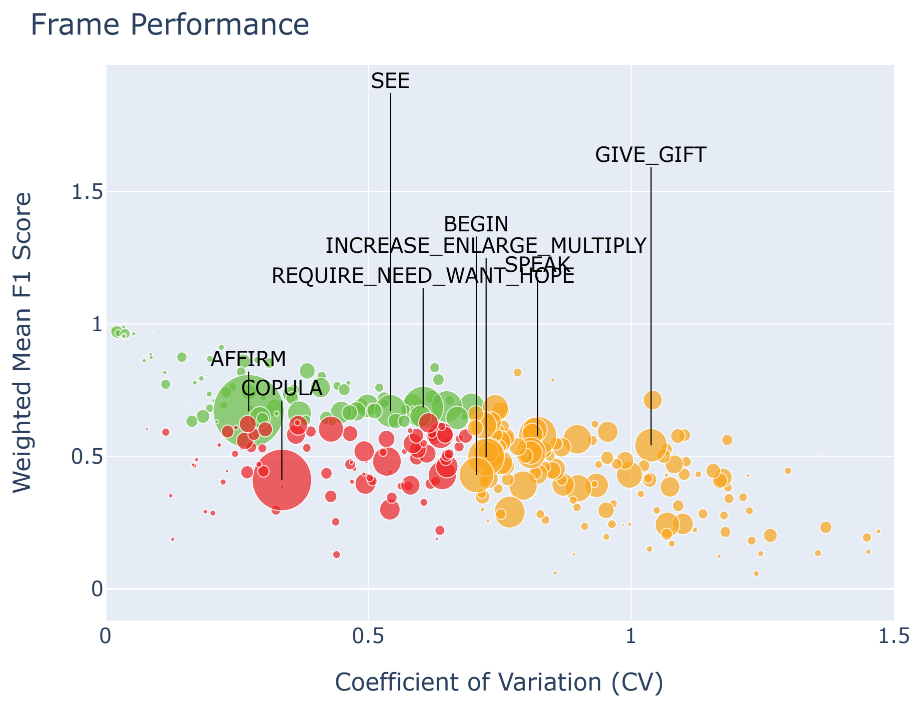

Figure 3: Scatter plot illustrating frame-level performance based on the weighted mean F1 score versus the coefficient of variation (CV). Point sizes correspond to the support value of each frame. Annotation added to frames with the highest support values.

the support value. This ensures a balanced representation of the grammars' performance across various frames. The main point of this plot is to provide a general insight into the grammars' performance on a frame level. It identifies specific frames where grammars excel or underperform. Additionally, the analysis can reveal commonalities or differences between frames, highlighting areas where grammars may require additional refinement.

Analyzing frames with high support values is essential for statistical significance and practical relevance. High support values result in more reliable and statistically significant results due to a larger sample size. From a practical standpoint, users are more likely to encounter frames with higher support values in real-world applications.

The plot groups the frames into three categories: best-performing, worst-performing, and high-variance frames, divided by the CV value with a cut-off point set to 0.7. The top half of consistently performing frames are labeled as "Consistently Higher Performing Frames" (e.g., AFFIRM), the bottom half as "Consistently Lower Performing Frames" (e.g., COPULA), and frames with high variability as "High Variance Frames" (e.g., SPEAK). Colors differentiate these groups, and the top eight frames with the largest support values are annotated with text labels.

Given the results from the macro evaluation, it is expected that the **Frequency-Based Priority**, **Network Connection Strength**, and **Local Dependency Preference** heuristics have a significant influence on the performance of the grammar, especially on frames with a high support value. This is confirmed by examining the CPS for the "AFFIRM" frame (Figure 4). As described in the methodology section, the CPS is a synthesized score derived from the differences in F1 scores between the active

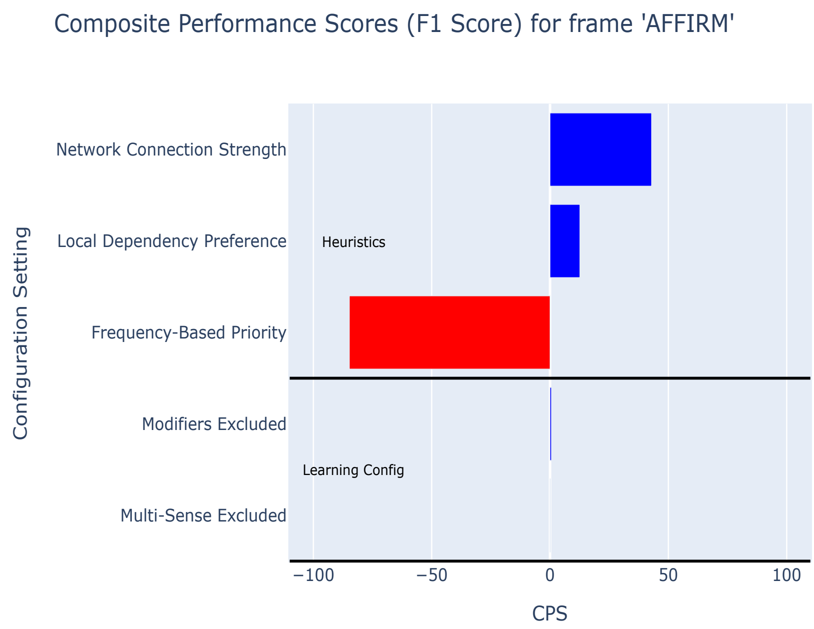

Figure 4: Plot showing CPS results for frame "AFFIRM" (metric F1). Blue and red bars represent positive and negative CPS values, respectively. A positive CPS indicates that enabling the specific configuration resulted in higher F1 scores for this specific frame.

and inactive states of configuration settings. A positive CPS indicates that enabling a specific configuration resulted in higher F1 scores for this frame, while a negative CPS suggests that the configuration led to lower F1 scores. The "AFFIRM" frame has the highest support value. It is described as "An agent AFFIRMS a theme to a recipient (+attribute)" (Di Fabio, Conia, and Navigli 2019). It is composed of the following PropBank role sets: "say.01," "claim.01," "insist.01," "allege.01," "confirm.01," "assert.03," and "contend.01". The role set "say.01" has the highest support value of 109,851, with a mean F1-score of 0.66. Other role sets in this frame have a much lower support value.

There are positive scores for **Network Connection Strength** (CPS: 42.9) and **Local Dependency Preference** (CPS: 12.6), indicating that the grammars perform better when one of these heuristics is used. On the other hand, the **Frequency-Based Priority** (CPS: -8.48) greatly negatively impacts the grammars' performance. In other words, the CPS table for the frame "AFFIRM" aligns with the macro CPS plot, indicating that the performance of the grammars on the "AFFIRM" frame is consistent with their overall performance. Interestingly, the learning configurations did not impact the grammars' performance on the "AFFIRM" frame.

To examine the frame performance more thoroughly, this study analyzed the specific CPS for grammars using these heuristics to determine which heuristics perform best on which frames (Figure 5). The focus is on frames where a specific heuristic significantly influences the F1 score, as determined by the CPS. A frame was included in this analysis if its CPS was higher than 10 and it was the highest score compared to the other heuristics. Identifying such frames helps researchers understand which

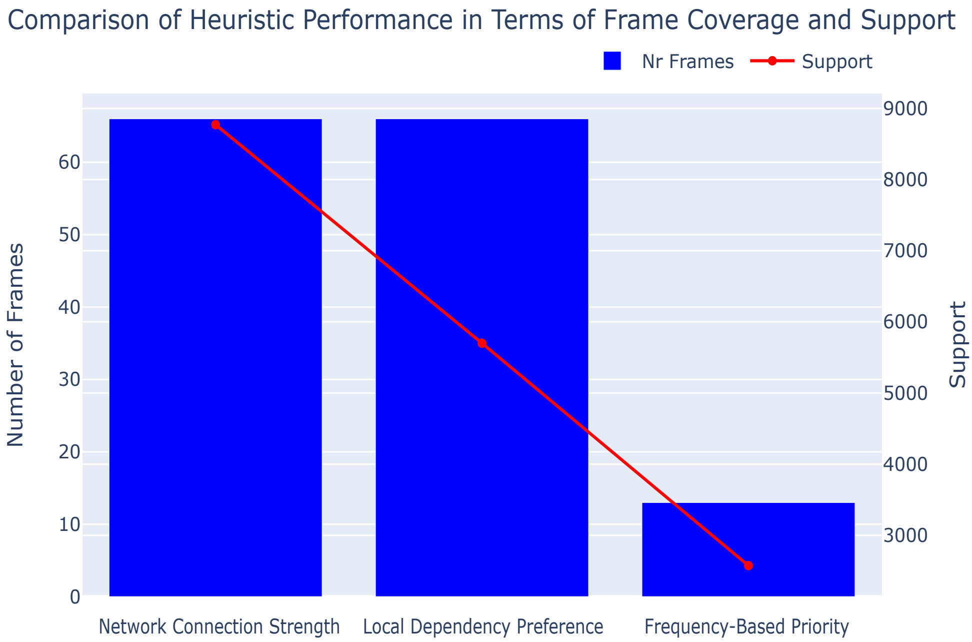

Figure 5: Comparison of Heuristic Performance in Terms of Frame Coverage and Support. The x-axis shows the three main heuristics: Network Connection Strength, Local Dependency Preference, and Frequency-Based Priority. The left y-axis (blue bars) represents the number of frames where each heuristic had a large impact (CPS > 10) and outperformed others. The right y-axis (orange dots) shows the mean support value of these frames. This plot illustrates both the breadth (number of frames) and depth (average support) of each heuristic's effectiveness.

heuristics perform exceptionally well in specific instances, informing potential areas for optimization and further development.

Figure 5 provides an overview of how each heuristic performs. The x-axis displays the three main heuristics: Network Connection Strength, Local Dependency Preference, and Frequency-Based Priority. For each heuristic, two key metrics are shown: (1) the blue bars, corresponding to the left y-axis, represent the number of frames where the heuristic had a large impact (CPS > 10) and outperformed the others. This metric indicates the breadth of the heuristic's effectiveness across frames. (2) The red dots, corresponding to the right y-axis, show the mean support value of these frames. The support value reflects how frequently these frames appear in the dataset.

The **Network Connection Strength** and **Local Dependency Preference** heuristics have a significantly higher number of frames where they outperform the other heuristics. Moreover, they also perform better on frames with higher support values. This aligns with the findings from the macro evaluation. On the other hand, the other heuristic (**Frequency-Based Priority**) has significantly fewer frames where they score the best, and these frames also have a lower support value overall.

When analyzing the frame performance in relation to specific heuristics (Figure 6), it is evident that specific heuristics have a more considerable positive impact on particular frames. Investigating these relationships between heuristic settings and grammar performance across various frame contexts can provide valuable insights to develop

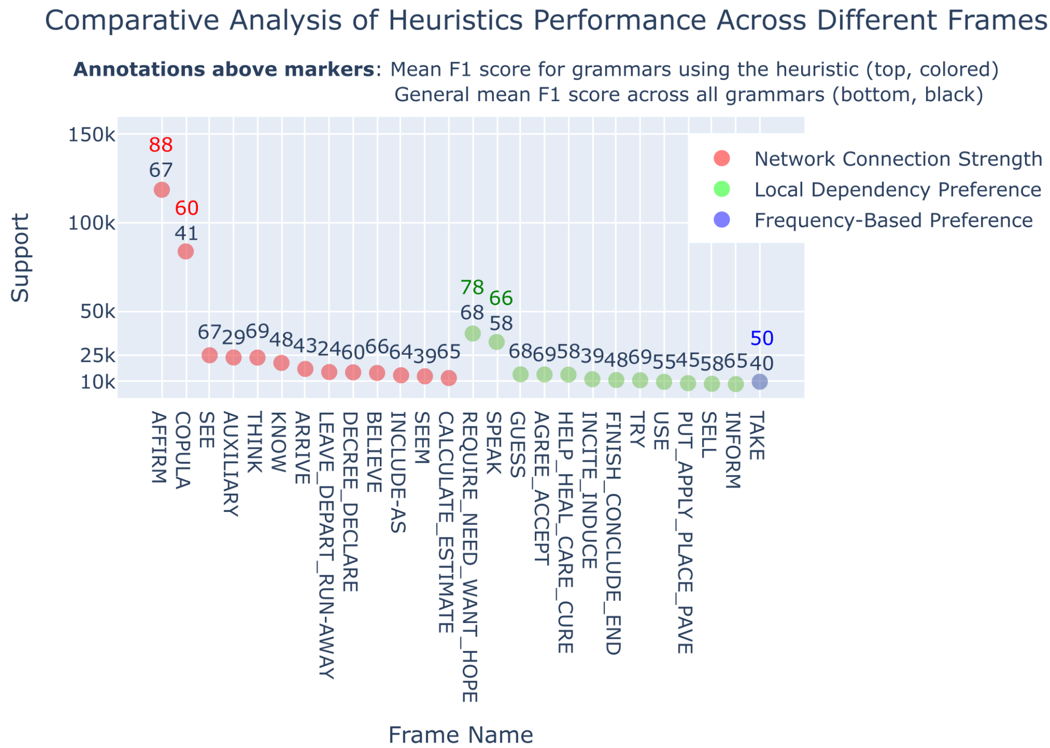

Figure 6: Visualization of heuristics' performance by frame, highlighting which heuristic achieved the highest F1 score for each frame. Different colors represent distinct heuristics, while the numbers indicate each mean F1 score across all grammars (bottom), and for the highest support frames, an extra number shows the mean F1 score for grammars using the top-performing heuristic (top).

more effective grammar configurations.

For frames with high support values, the **Network Connection Strength** heuristic stands out in its performance. For example, in the case of the "AFFIRM" frame, which has a support of 117204, the **Network Connection Strength** heuristic has a composite performance score (CPS) of 43.3, indicating that grammars using this heuristic score significantly higher than the general F1 score of 0.67. Similarly, for the "COPULA" frame with a support of 82296, the **Network Connection Strength** heuristic has a CPS of 37.8. These high CPS values suggest that the **Network Connection Strength** heuristic significantly improves the grammars' performance for these frames.

Similarly, the **Local Dependency Preference** heuristic performs well for specific frames. For the "REQUIRE_NEED_WANT_HOPE" frame with a support of 38818, the **Local Dependency Preference** heuristic has a CPS of 19.9. The same trend is observed for the "SPEAK" frame, with a support of 34042, where the **Local Dependency Preference** heuristic has a CPS of 15.8. These findings suggest that the **Local Dependency Preference** heuristic helps the grammars to more accurately capture the relationships between different elements within these frames, resulting in improved performance.

The **Frequency-Based Priority** heuristic shows a significant positive impact on the "TAKE" frame, with a CPS of 18.0 compared to the general F1 score of 0.40. This indicates that the **Frequency-Based Priority** heuristic helps the grammars to better handle the linguistic patterns within this frame, resulting in improved performance. However, as expected based on the macro evaluation, this heuristic hurts the overall performance of the grammars. This is because the **Frequency-Based Priority** heuristic performs poorly on most frames with high support values.

5 Discussion and Future Directions

This discussion examines the findings and their potential implications for the ongoing development of large-scale construction grammars. The insights from the macro and frame-level evaluations and analyses of construction applications across different frame clusters provide a nuanced understanding of how various grammar configurations perform in predicting different PropBank frames. The analysis highlights that heuristics play an important role. Specifically, the three heuristics examined, **Frequency-Based Priority**, **Local Dependency Preference**, and **Network Connection Strength** have a significant impact on grammar performance, suggesting that the implementation of well-designed heuristics is crucial.

The consistent performance of the **Network Connection Strength** heuristic across various frames and configurations indicates that the connections between constructions in the categorial network are relevant for accurate frame prediction. This heuristic's effectiveness suggests that the strength of associations between constructions is a factor in language processing, potentially reflecting aspects of how linguistic knowledge is accessed and utilized during comprehension. The performance of this heuristic in predicting frames suggests that the network structure of linguistic knowledge may be more than a representational choice but a fundamental aspect that directly influences language processing.

Examining frame performance revealed specific trends and variations across different verb clusters, which provide insights into the nature of different semantic frames and their processing requirements. In the "AFFIRM" frame (e.g., "Corporate lawyers said the new fees wouldn't inhibit many"), the **Network Connection Strength** heuristic performs well. The construction application process for this heuristic involved navigating complex relationships between multiple entities and actions. It correctly identified "said" as the main predicate with the sense "say.01" (meaning "to express in words") and accurately assigned the arguments "Corporate lawyers" as Arg0 and the entire clause "the new fees wouldn't inhibit many" as Arg1. This success might be attributed to the heuristic's ability to leverage the interconnected nature of the categorial network, allowing it to handle complex, embedded structures effectively. In contrast, the **Frequency-Based Priority** heuristic struggled with this sentence, failing to capture the embedded structure. It only identified "said" as the main predicate without correctly assigning the complex Arg1. This limitation could be due to its reliance on frequency data, which may not adequately represent the nuanced relationships in complex sentences. The **Local Dependency Preference** heuristic performed better than **Frequency-Based Priority** but still missed some nuances, incorrectly identifying "lawyers" as Arg0 without including "Corporate". This partial success might stem from its focus on nearby syntactic relationships, which captures some but not all of the sentence's complexity with this particular semantic frame.

For the “REQUIRE_NEED_WANT_HOPE” frame (e.g., “They want to tell the Good News in the areas”), the **Local Dependency Preference** heuristic shows good performance. In processing this sentence, the heuristic prioritized local syntactic relationships, correctly identifying “want” with the sense “want.01” (meaning “to desire”) as the main predicate and “to tell the Good News in the areas” as its complement. This success may be due to the heuristic’s emphasis on nearby syntactic relationships, which aligns well with the structure of desire expressions in English. The **Frequency-Based Priority** heuristic struggled, failing to capture the relationship between “want” and the infinitive clause “to tell”. It only identified “want” and “tell” as separate predicates without connecting them properly.

Interestingly, in the “TAKE” frame (e.g., “She took her business to First Atlanta”), the **Frequency-Based Priority** heuristic outperforms others, matching the gold standard perfectly. This heuristic correctly identified “took” with the sense “take.01” (meaning “to move something or someone”) as the main predicate and accurately assigned “She” as Arg0, “her business” as Arg1, and “to First Atlanta” as Argm-dir. This success might be due to the high frequency and relatively straightforward structure of such “take” expressions in the training data. Surprisingly, the **Network Connection Strength** heuristic underperformed in this example, misidentifying the frame and failing to assign core arguments correctly. This unexpected failure could be due to an overemphasis on certain network connections, leading to misinterpreting the verb’s role. The **Local Dependency Preference** heuristic performed better than **Network Connection Strength** but still misclassified “to First Atlanta” as Arg3 instead of Argm-dir. This partial success might be attributed to its accurate capture of local relationships but a failure to distinguish between core and modifier arguments in this context.

These variations across verb clusters reflect the complexity of verb semantics and their associated frames, suggesting that verbs and their frames may have specific structural and semantic properties that affect how they are processed. This finding has implications for both understanding language structure and computational implementations. It suggests that different semantic domains might be structured and processed differently. Some concepts might be organized more in terms of their frequency and common usage, while others might rely more on complex networks of associations or local syntactic patterns. It indicates that a one-size-fits-all approach to heuristics in construction grammars may not be optimal for computational implementations. Instead, future development of computational construction grammars could consider a more nuanced, frame-specific approach to heuristic application. This could involve developing mechanisms for heuristic selection or weighting based on frame properties. For instance, for frames like “AFFIRM” that involve complex relationships, the **Network Connection Strength** heuristic could be given higher weight. For frames expressing desires or intentions like “REQUIRE_NEED_WANT_HOPE”, the **Local Dependency Preference** heuristic might be prioritized. For common action frames like “TAKE”, the **Frequency-Based Priority** heuristic could play a more significant role. Additionally, exploring ways to combine heuristics meaningfully could lead to more robust and flexible grammar implementations. For example, a weighted combination of **Network Connection Strength** and **Frequency-Based Priority** might capture both the structural complexity and the common usage patterns of specific frames.

Moreover, exploring how these computational findings relate to psycholinguistic research on human language processing could lead to valuable insights.

The challenges observed in handling modifiers and multi-sense verb rolesets indicate areas for potential improvement. Future research should explore techniques for better representing and processing these complex linguistic elements within the framework. This might include developing specialized processing strategies for modifiers and ambiguous verbs, potentially informed by studies on human language comprehension. The insights from this study suggest that further alignment between computational models and linguistic theory could be beneficial. For instance, the performance of network-based heuristics could inform theoretical models of language acquisition and processing. Similarly, insights from psycholinguistic research on semantic frame processing could guide the development of computational models.

The practical utility of CxG in applications like frame-semantic analysis of texts and construction-based corpus searches has been suggested by the findings. However, this potential has not yet been fully realized. In light of the findings, it is evident that developing more refined and complete heuristics is crucial to enhancing the performance of construction grammars. A more qualitative approach could also be employed to delve deeper into the workings of the heuristics. Examining specific cases where heuristics succeed or fail in capturing the nuanced relationships between different sentence elements could offer more concrete insights into their mechanics and effectiveness. Such an approach would allow pinpointing the areas where improvements are needed and guide the development of more precise and robust heuristics.

Moreover, an intriguing avenue for future research would be to explore the potential application of these grammars in the formulation of sentences, as opposed to comprehension. This would involve utilizing the grammars to formulate a sentence from a given meaning representation, thereby reversing the current process of comprehending a sentence to derive its meaning representation. Investigating the efficacy of construction grammars in this context could provide valuable insights into their versatility and practical utility in various language applications.

6 Conclusion

This study provides an evaluation of large-scale, computational construction grammars and their ability to extract semantic roles and frames from English sentences. The analysis of the computational construction grammars has offered valuable insights that contribute to the refinement and progression of future grammars. These findings highlight the potential and importance of construction grammars in the fields of natural language processing and artificial intelligence.

The study identifies **Frequency-Based Priority**, **Local Dependency Preference**, and **Network Connection Strength** as key heuristics in defining the search and application process for extracting semantic frames. The importance of these heuristics in affecting the performance of grammars is evident. However, all grammars examined faced challenges in specific areas, particularly in handling modifiers and multi-sense verb rolesets, providing clear directions for future research and development.

The study reveals complex relationships between heuristics and specific semantic frames, showing that the effectiveness of certain heuristics varies depending on the frames and their semantic and syntactic information. For instance, **Network Connection Strength** performs well in frames with higher support like "AFFIRM" and "SEE", while **Local Dependency Preference** excels in frames like "REQUIRE_NEED_WANT_HOPE" and "SPEAK". Interestingly, **Frequency-Based Priority**, while generally underperforming, shows unexpected strength in the "TAKE" frame.

These variations suggest that different semantic domains might be structured and processed differently, indicating that a one-size-fits-all approach to heuristics in construction grammars may not be optimal. Instead, future development of computational construction grammars could benefit from a more nuanced, frame-specific approach to heuristic application.

The potential practical utility of Construction Grammar in applications such as frame-semantic analysis of texts and construction-based corpus searches is evident (see also Beuls, Van Eecke, and Cangalovic 2021). Still, this potential has yet to be fully realized. Developing more refined and complete heuristics is crucial to enhancing the performance of construction grammars, particularly when dealing with modifiers and certain rolesets.

Future research directions include exploring hybrid approaches that combine different heuristics, developing adaptive systems for heuristic selection based on frame properties, and investigating the application of these grammars in sentence formulation. Additionally, addressing the challenges in handling modifiers and multi-sense verb rolesets could significantly improve the overall performance of these grammars.

As integration between Cognitive Linguistics and AI progresses, construction grammars could play a significant role. Insights from this study could inform the refinement of heuristics and encourage deeper exploration of construction grammars in linguistic and AI tasks. The varying effectiveness of different heuristics across frame types suggests a complex picture of language processing, pointing towards a model where different semantic domains might be structured and accessed in different ways.

There is still much to uncover about the capabilities of construction grammars. As these computational models continue to be refined, they may offer deeper insights into the nature of language and cognition. The future holds numerous opportunities for further research and innovation in this intriguing and continuously evolving field.

References

Barres, Victor J. 2017. "Template Construction Grammar: A Schema-Theoretic Computational Construction Grammar." In *2017 AAAI Spring Symposium Series*.

Bergen, Benjamin K, and Nancy Chang. 2005. "Embodied Construction Grammar in Simulation-Based Language Understanding." *Construction Grammars: Cognitive Grounding and Theoretical Extensions. John Benjamins*.

Beuls, Katrien, 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)*, 41–50. https://aclanthology.org/2023.cxgsnlp-1.6/.

———. 2024. "Construction Grammar and Artificial Intelligence." In *The Cam-*

bridge Handbook of Construction Grammar, edited by Mirjam Fried and Kiki Nikiforidou. Cambridge: Cambridge University Press.

Beuls, Katrien, Paul Van Eecke, and Vanja Sophie Cangalovic. 2021. "A Computational Construction Grammar Approach to Semantic Frame Extraction." *Linguistics Vanguard* 7 (1). https://doi.org/10.1515/lingvan-2018-0015.

Beuls, Katrien, and Remi van Trijp. 2016. "Computational Construction Grammar and Constructional Change." *Belgian Journal of Linguistics* 30 (1): 1–13. https://doi.org/10.1075/bjl.30.01beu.

Boas, Hans C. 2021. "Construction Grammar and Frame Semantics." In *The Routledge Handbook of Cognitive Linguistics*, 1st ed., 35. Routledge.

Bybee, Joan L. 2006. "From Usage to Grammar: The Mind's Response to Repetition." *Language* 82 (4): 711–33.

Croft, William. 2001. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford University Press on Demand.

Croft, William, and Alan Cruse. 2004. *Cognitive Linguistics*. Cambridge University Press.

Di Fabio, Andrea, Simone Conia, and Roberto Navigli. 2019. "VerbAtlas: A Novel Large-Scale Verbal Semantic Resource and Its Application to Semantic Role Labeling." In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 627–37. Hong Kong, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1058.

Diessel, Holger. 2019. *The Grammar Network*. Cambridge University Press.

———. 2023. *The Constructicon: Taxonomies and Networks*. Elements in Construction Grammar. Cambridge University Press. https://doi.org/10.1017/9781009327848.

Eisenstein, Jacob. 2019. *Introduction to Natural Language Processing*. MIT press.

Fillmore, Charles. 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*, 280:20–32. 1. New York.

———. 1985. "Syntactic Intrusions and the Notion of Grammatical Construction." In *Annual Meeting of the Berkeley Linguistics Society*, 11:73–86.

———. 2006. "Chapter 10 Frame Semantics." In *Cognitive Linguistics: Basic Readings*, edited by Dirk Geeraerts, 373–400. Berlin, New York: De Gruyter Mouton. https://doi.org/doi:10.1515/9783110199901.373.

Fillmore, Charles, Paul Kay, and Mary O'Connor. 1988. "Regularity and Idiomaticity in Grammatical Constructions: The Case of Let Alone." *Language* 64 (3): 501–38. https://doi.org/10.2307/414531.

Geeraerts, Dirk. 2006. *Cognitive Linguistics: Basic Readings*. Vol. 34. Walter de Gruyter.

Gibson, Edward. 2000. "The Dependency Locality Theory: A Distance-Based Theory of Linguistic Complexity." In *Image, Language, Brain: Papers from the First Mind Articulation Project Symposium*, 95–126. MIT Press.

Goffman, Erving. 1974. *Frame Analysis: An Essay on the Organization of Experience.* Harvard University Press.

Goldberg, Adele. 1995. *Constructions: A Construction Grammar Approach to Argument Structure*. The University of Chicago Press.

———. 2006. *Constructions at Work: The Nature of Generalization in Language*. Constructions at Work: The Nature of Generalization in Language. Oxford University Press. https://books.google.be/books?id=LHrcqeZmUN4C.

Haspelmath, Martin. 2023. "On What a Construction Is." *Constructions* 15 (1). https://doi.org/10.24338/cons-539.

He, Luheng, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018a. "Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling." *arXiv Preprint arXiv:1805.04787*.

———. 2018b. "Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling." In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, edited by Iryna Gurevych and Yusuke Miyao, 364–69. Melbourne, Australia: Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-2058.

Hoffmann, Thomas. 2017. "Construction Grammars." In *The Cambridge Handbook of Cognitive Linguistics*, 310–29. https://doi.org/10.1017/9781316339732.020.

———. 2022. *Construction Grammar: The Structure of English*. Cambridge University Press.

Jurafsky, Daniel, and James Martin. 2020. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Vol. 2.

Kay, Paul. 1990. "Even." *Linguistics and Philosophy* 13 (1): 59–111. https://doi.org/10.1007/BF00630517.

Li, Zuchao, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019. "Dependency or Span, End-to-End Uniform Semantic Role Labeling." In *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:6730–37. 01.

Màrquez, Lluís, Xavier Carreras, Kenneth Litkowski, and Suzanne Stevenson. 2008. "Semantic Role Labeling: An Introduction to the Special Issue." *Computational Linguistics* 34 (June): 145–59. https://doi.org/10.1162/coli.2008.34.2.145.

Mickus, Timothee, Denis Paperno, Mathieu Constant, and Kees Van Deemter. 2019. "What Do You Mean, BERT? Assessing BERT as a Distributional Semantics Model." *arXiv Preprint arXiv:1911.05758*.

Minsky, Marvin. 1974. "A Framework for Representing Knowledge." MIT, Cambridge.

Nevens, Jens, Paul Van Eecke, and Katrien Beuls. 2019. "Computational Construction Grammar for Visual Question Answering." *Linguistics Vanguard* 5 (1). https://doi.org/10.1515/lingvan-2018-0070.

Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. "The Proposition Bank: An Annotated Corpus of Semantic Roles." *Computational Linguistics* 31 (1): 71–106. https://doi.org/10.1162/0891201053630264.

Pradhan, Sameer, 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*, 278–88. Seattle, Washington: Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.starsem-1.24.

Pradhan, Sameer, Wayne Ward, Kadri Hacioglu, James H Martin, and Dan Jurafsky. 2005. "Semantic Role Labeling Using Different Syntactic Views." In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, 581–88.

Reimers, Nils, and Iryna Gurevych. 2019. "Sentence-Bert: Sentence Embeddings Using Siamese Bert-Networks." *arXiv Preprint arXiv:1908.10084*.

Ruppenhofer, Josef, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. "FrameNet II: Extended Theory and Practice." In.

Russell, Stuart, and Peter Norvig. 2021. *Artificial Intelligence: A Modern Approach*. 4th ed. Pearson Education.

Steels, Luc. 2011. *Design Patterns in Fluid Construction Grammar*. Vol. 11. John Benjamins Publishing.

———. 2017. "Basics of Fluid Construction Grammar." *Constructions and Frames* 9 (2): 178–225. https://doi.org/10.1075/cf.00002.ste.

Surdeanu, Mihai, Lluı□s Màrquez, Xavier Carreras, and Pere R Comas. 2007. "Combination Strategies for Semantic Role Labeling." *Journal of Artificial Intelligence Research* 29: 105–51.

Ungerer, Tobias, and Stefan Hartmann. 2023. "Constructionist Approaches: Past, Present, Future." https://doi.org/10.31234/osf.io/83dvj.

Van Eecke, Paul. 2018. "Generalisation and Specialisation Operators for Computational Construction Grammar and Their Application in Evolutionary Linguistics Research."

Van Eecke, Paul, and Katrien Beuls. 2018. "Exploring the Creative Potential of Computational Construction Grammar." *Zeitschrift Für Anglistik Und Amerikanistik* 66 (3): 341–55. https://doi.org/10.1515/zaa-2018-0029.

Van Eecke, Paul, Jens Nevens, and Katrien Beuls. 2022. "Neural Heuristics for Scaling Constructional Language Processing" 10 (December): 287–314. https://doi.org/10.15398/jlm.v10i2.318.

van Trijp, Remi. 2017. "A Computational Construction Grammar for English." In *2017 AAAI Spring Symposium Series*, 266–73. Palo Alto, CA: AAAI Press.

———. To appear. "Different Constructional Approaches in Practice: A Comparative Study." In *Cambridge Handbook of Construction Grammar*. Cambridge: Cambridge University Press.

van Trijp, Remi, Katrien Beuls, and Paul Van Eecke. 2022. "The FCG Editor: An Innovative Environment for Engineering Computational Construction Grammars." Edited by Andrew Kehler. *PLOS ONE* 17 (6): e0269708. https://doi.org/10.1371/journal.pone.0269708.

Weischedel, Ralph, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, and Michelle Franchini. 2013. "Ontonotes Release 5.0." *Linguistic Data Consortium, Philadelphia, PA* 23.

Appendices

A Learning PropBank Grammars

The process of learning large-scale construction grammars from PropBank annotations has been implemented within the Babel environment[6]. This environment supports a wide range of operations that pertain to language processing, learning, and analysis. The procedure of deriving a construction grammar from a PropBank annotated corpus adheres to the following stages:

1. It begins with a set of sentences, each with a PropBank annotation layer, serving as the input.
2. Constructions, which are pairings of form and meaning, are extracted from these annotated sentences according to specified learning configurations. In this context, the *form* refers to syntactic tree structures, while the *meaning* corresponds to semantic roles. The learning configurations play a crucial role in selecting or excluding certain rolesets and determining which roles will be included in the learning phase.
3. The end result is a collection of newly learned constructions that are compiled into a construction inventory.

After completing this procedure for the entire set of sentences, the resulting construction inventory, along with its associated categorial network, forms the grammar. The construction inventory contains the learned constructions, while the categorial network organizes these constructions into a systematic and interconnected structure. As more constructions are learned from annotated PropBank sentences, both the inventory and the categorial network continue to expand and diversify. Together, they can then be used to comprehend and extract frames from unannotated sentences. This learning process effectively creates a PropBank grammar based on a corpus of PropBank-annotated sentences, enabling the operationalization of large-scale usage-based construction grammar.

To provide a clearer understanding of how this categorial network evolves as it integrates more learned constructions, a practical illustration is given. This example will depict the development of the categorial network, showcasing how the learning and incorporation of new constructions incrementally expand this network. Additionally, it highlights the features of the acquired constructions.

This practical example involves the creation of a custom corpus including PropBank annotated sentences. First, the PropBank annotated sentence(s) added to the corpus will be shown. This is then followed by the state of the construction inventory and categorial network after the grammar has learned and incorporated the constructions from these sentences.

The first annotated sentence that will be used to learn constructions is "I tell my

[6] https://emergent-languages.org/

sister a story" (Table 5).

Table 5: PropBank annotations for "I tell my sister a story"

String	Sense	Lemma	Roles
I	-	-	ARG0
tell	tell.01	tell	V
my sister	-	-	ARG2
a story	-	-	ARG1

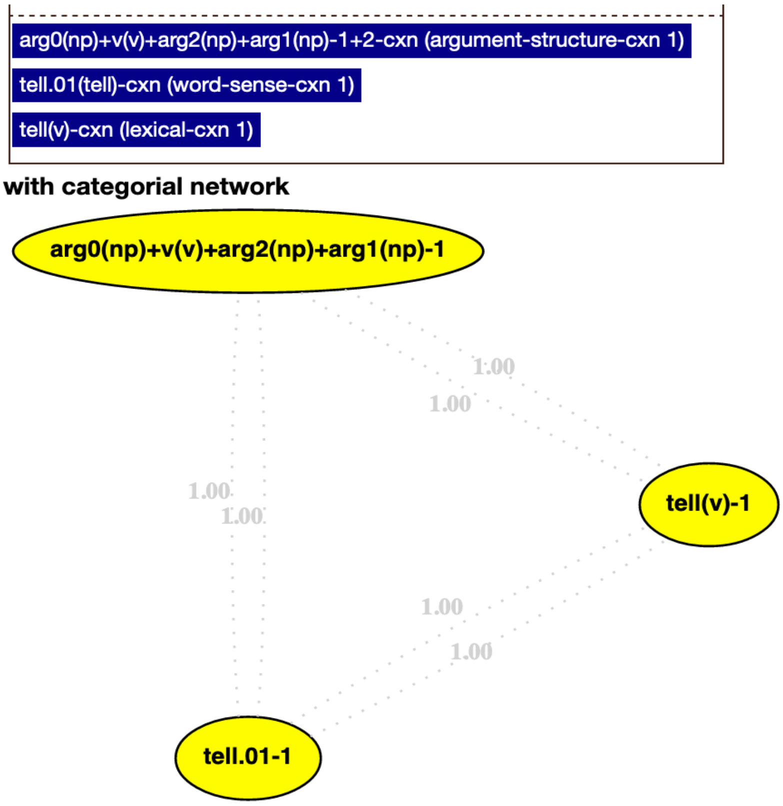

Figure 7: Construction inventory and categorial network after learning "I tell my sister a story"

As the construction inventory and categorial network show in Figure 7, from the sentence "I tell my sister a story", three constructions were learned. The lexical construction "tell(v)" (Figure 8), the word sense construction "tell.01" (Figure 9) and the argument structure construction "arg0(np)-v(v)-arg2(np)-arg1(np)" (Figure 10). Specifically, lexical constructions pinpoint the verb's form, word sense constructions represent the verb's context or specific sense, and argument structure constructions bridge syntactic structures to semantic roles at precise locations within the syntactic tree. These are connected to each other in a network (Figure 7).

Zooming in on the learned argument structure construction (Figure 10), several components are present:

- **arg0(np)+v(v)+arg2(np)+arg1(np)-1+2-cxn (argument-structure-cxn 1)**: This identifier represents a complex argument structure construction. It indicates that this construction is concerned with a sentence structure involving an ARG0

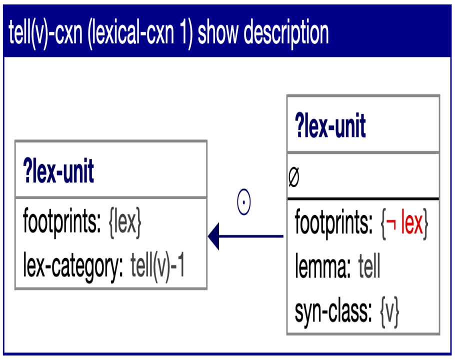

Figure 8: The lexical construction for the verb "tell". It shows the unit's designation as a verb (**{v}**) with the lemma "tell" and its category within the grammar as **tell(v)-1**. This construction is used to identify and process the verb "tell".

noun phrase (np), a verb (v), an ARG2 noun phrase, and an ARG1 noun phrase.

Moving to the left box:

- **?tell-1**: This unit within the construction contributes to the overall meaning and structure of the construction.
  - **frame-evoking**: It indicates the construction is responsible for evoking a semantic frame, providing the structure for a situation that includes various participants and their actions.
  - **footprints: {fee}**: It indicates that a frame-evoking element construction has already been applied in the comprehension process.
  - **gram-category**: It specifies the grammatical categories from the categorial network that are involved in this construction, corresponding to ARG0 (subject noun phrase), the verb, ARG2 (indirect object noun phrase), and ARG1 (direct object noun phrase).
  - **frame**: This is a placeholder for the name of the semantic frame that the construction is part of.
  - **meaning**: It describes the semantic roles associated with the frame elements, indicating the construction's role in mapping the syntactic structure to the semantic roles within the specified frame.

On the right side of the figure, several units represent different components of the sentence structure that fit this construction. Each unit is connected to a parent unit, indicating its place in the hierarchy of the sentence structure. The syntactic class (syn-class) for each unit indicates whether it is a noun phrase (np), verb (v), sentence (s), or verb phrase (vp).

This construction, as a whole, captures how the verb "tell" is used in a sentence with a subject, indirect object, and direct object, and how these elements are interrelated both syntactically and semantically. More extensive coverage of how constructions are designed in FCG can be found in Chapter 3 of Van Eecke's (2018) dissertation.

Now, a second annotated sentence will be added: "I told a story to my sister" Table 6.

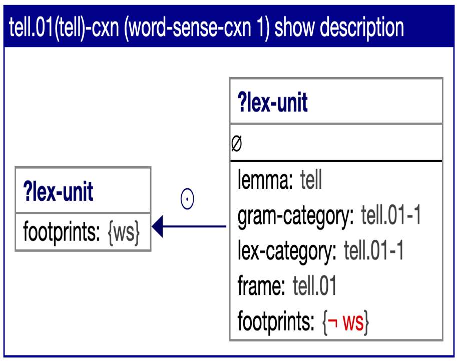

Figure 9: The word sense construction for the sense “tell.01”. This construction is crucial for distinguishing between different meanings of the verb “tell” in processing.

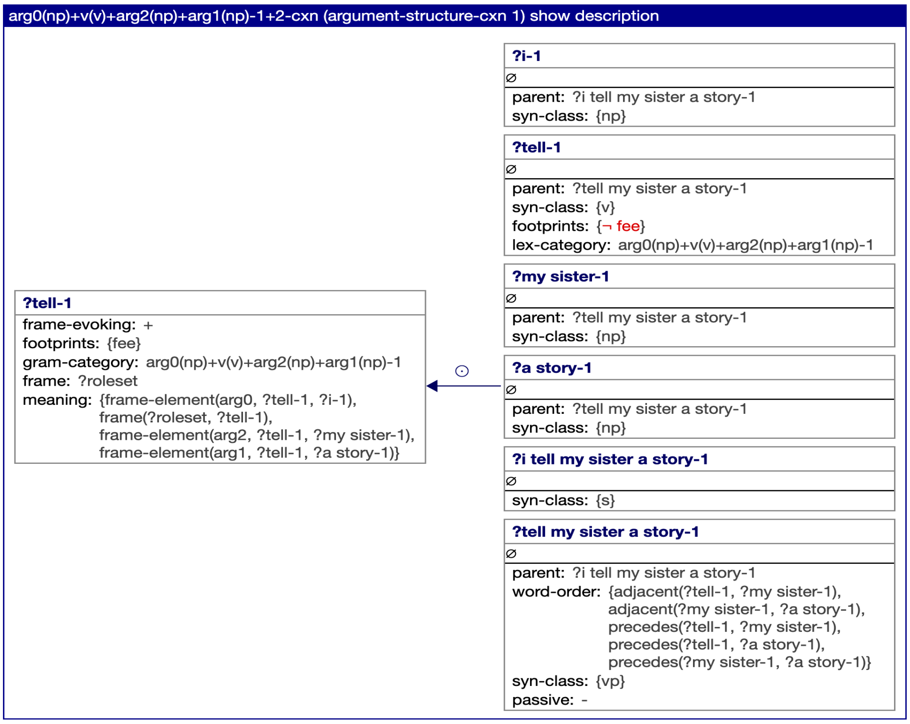

Figure 10: The argument structure construction extracted from the sentence “I tell my sister a story”.

Table 6: PropBank annotations for “I told a story to my sister”

String	Sense	Lemma	Roles
I	-	-	ARG0
told	tell.01	tell	V
a story	-	-	ARG1
to my sister	-	-	ARG2

This sentence contains the same word sense (tell.01) and lexical (tell) construction but a different argument structure construction (ditransitive).

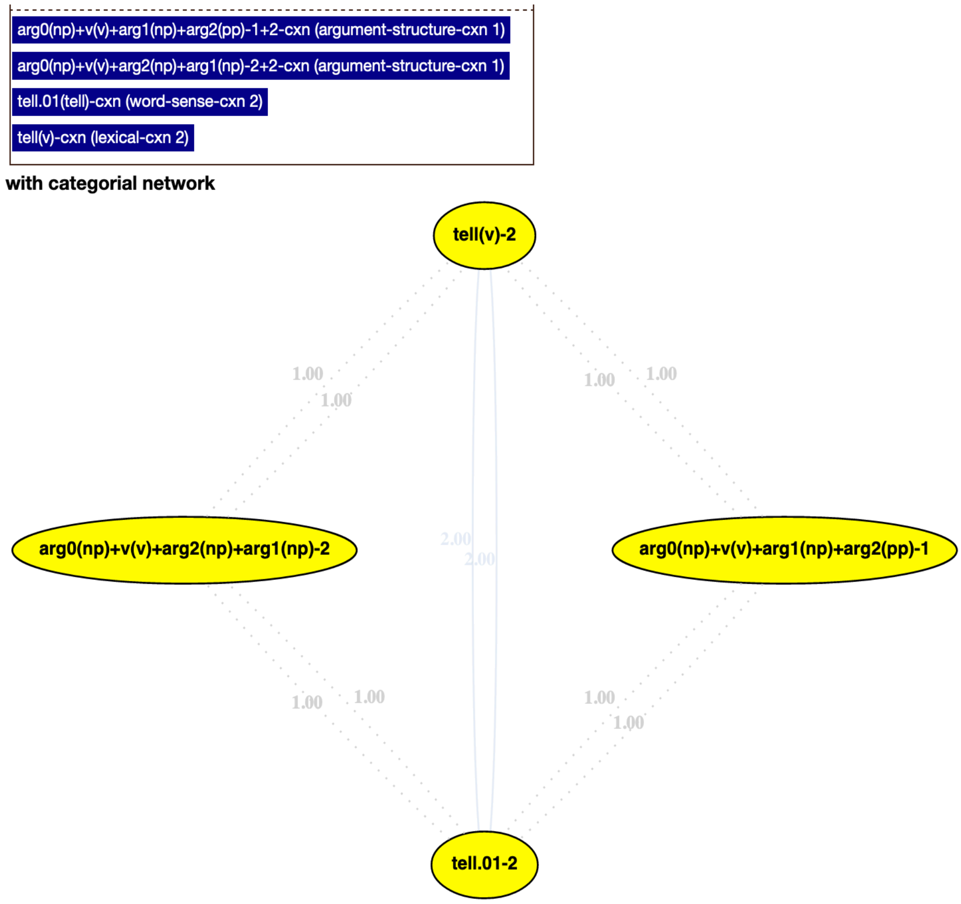

Figure 11: Construction inventory and categorial network after learning “I told a story to my sister”

Only one new construction is learned, namely the “arg0(np)-v(v)-arg1(np)-arg2(pp)” argument structure construction. This construction is added to the inventory and new links are made in the categorial network, connecting the existing word sense and lexical construction to the new argument structure construction.

Then, a couple more sentences are added to the corpus. These include the word sense “give.01” in both the ditransitive and dative alternation. In other words, no new argument structure constructions are added but a new lexical and word sense construction is brought into the corpus. The additional sentences are:

- I gave flowers to my mother (Table 7).
- I gave mother flowers (Table 8).
- She gives candy to her daughters (Table 9).

Table 7: PropBank annotations for “I gave flowers to my mother”

String	Sense	Lemma	Roles
I	-	-	ARG0
gave	give.01	give	V
flowers	-	-	ARG1
to, my mother	-	-	ARG2

Table 8: PropBank annotations for “I gave mother flowers”

String	Sense	Lemma	Roles
I	-	-	ARG0
gave	give.01	give	V
mother	-	-	ARG2
flowers	-	-	ARG1

Table 9: PropBank annotations for “She gives candy to her daughter”

String	Sense	Lemma	Roles
She	-	-	ARG0
gives	give.01	give	V
candy	-	-	ARG1
to her daughter	-	-	ARG2

By adding new word sense and lexical constructions, the network quickly expands. Now, not only “tell.01” and “tell(v) are connected to the ditransitive and dative alternation construction but these argument structure constructions are also linked to”give.01” and “give(v)”.

The next sentence that is added is “I send postcards to my mother” (Table 10).

Table 10: PropBank annotations for “I send postcards to my brother”

String	Sense	Lemma	Roles
I	-	-	ARG0
send	send.01	send	V
postcards	-	-	ARG1
to, my brother	-	-	ARG2

This sentence again features the same argument structure construction but a different word sense and lexical construction.

The word sense cxn “send.01” and lexical cxn “send(v)” are added to the inventory and network. In the categorial network, these nodes are linked to the ditransitive argument structure construction “arg0(np)-v(v)-arg1(np)-arg2(pp)” which in turn is linked to the other lexical and word sense constructions.

Next, the following sentences are added:

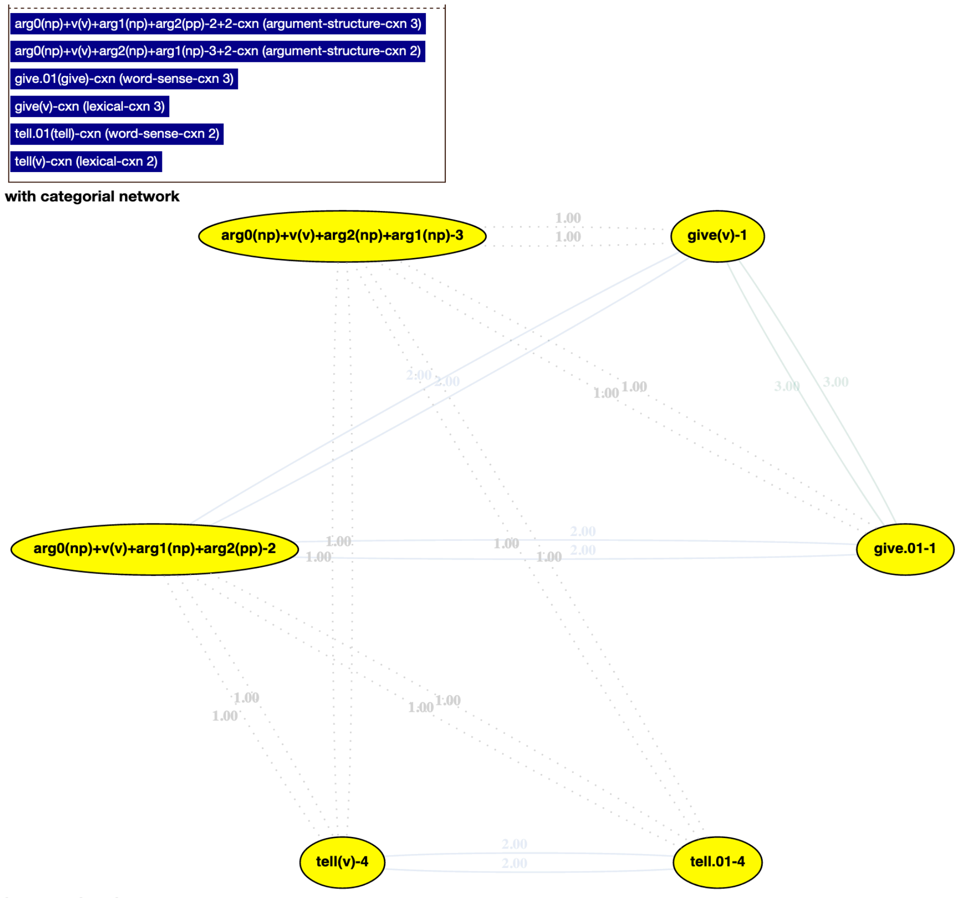

Figure 12: Construction inventory and categorial network after learning “give.01” and “give”

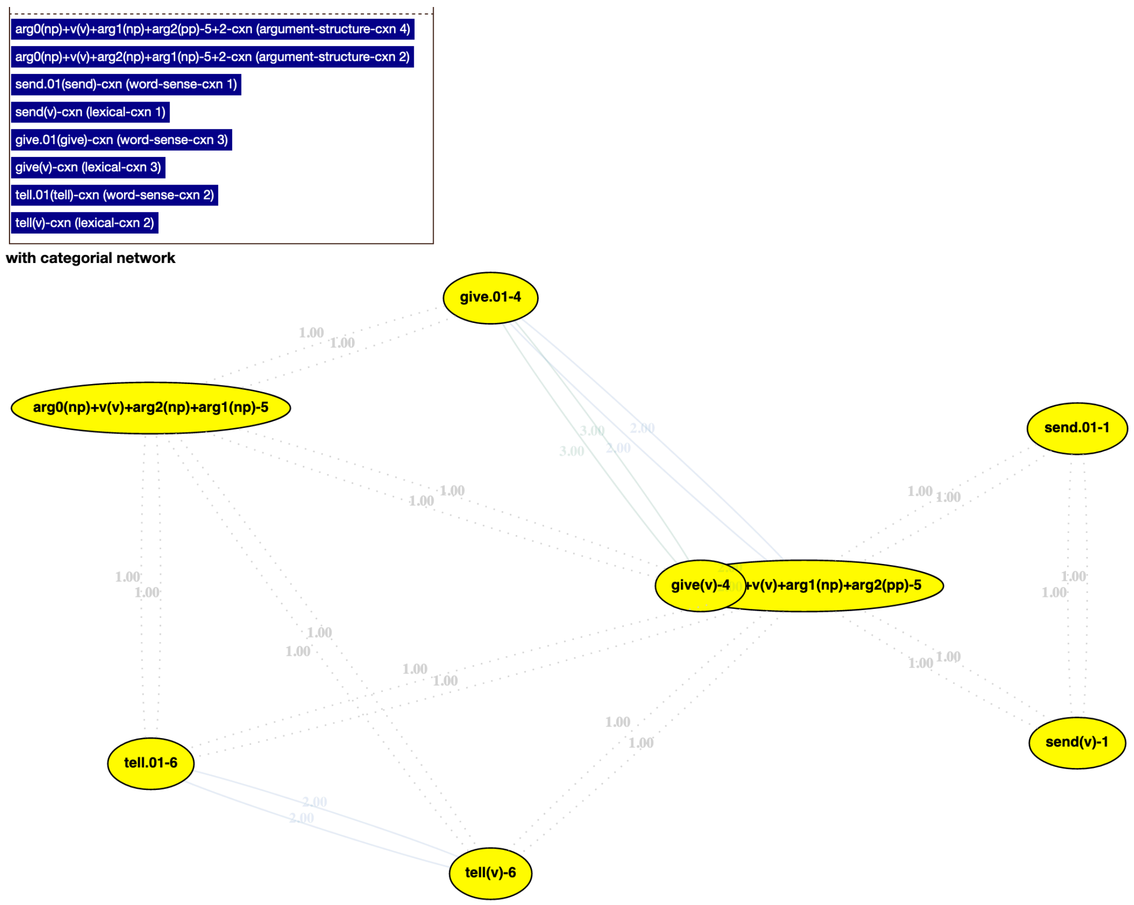

Figure 13: Construction inventory and categorial network after learning “send.01” and “send”

- He runs every morning (Table 11).
- I run the business (Table 12).
- I run a marathon (Table 13).
- I run in the park (Table 14).

Table 11: PropBank annotations for "He runs every morning"

String	Sense	Lemma	Roles
He	-	-	ARG0
runs	run.02	run	V
every morning	-	-	ARGM-TMP

Table 12: PropBank annotations for "I run the business"

String	Sense	Lemma	Roles
I	-	-	ARG0
run	run.01	run	V
the business	-	-	ARG1

Table 13: PropBank annotations for "I run a marathon"

String	Sense	Lemma	Roles
I	-	-	ARG0
run	run.02	run	V
a marathon	-	-	ARG1

Table 14: PropBank annotations for "I run in the park"

String	Sense	Lemma	Roles
I	-	-	ARG0
run	run.02	run	V
in the park	-	-	ARGM-LOC

These "run" sentences do not share anything with the previous annotated sentences. Therefore, they do not create any links to the other previously learned constructions. Specifically, they contain one new lexical construction (run), two new word sense constructions (run.01, run.02), two new argument structure constructions, a location modifier phrase construction (in the park) and a temporal modifier construction (every morning).

Note that only the location modifier phrase construction is added to the network. Additionally, the two word sense constructions share the argument structure construction "arg0(np)-v(v)-arg1(np)" indicating that these word senses can be embedded in this argument structure construction.

The final sentence includes a modal modifier: "We must prepare for the exams" (Table 15).

Table 15: PropBank annotations for "We must prepare for the exam"

String	Sense	Lemma	Roles
We	-	-	ARG0
must	-	-	ARGM-MOD
prepare	prepare.01	prepare	V
for the exam	-	-	ARG2

Once again, this new sentence can not be linked to any of the preexisting constructions in the network. They can only be linked to each other. It contains a new lexical construction (prepare), a new word sense construction (prepare.01), a new argument structure construction, and a new modal modifier construction (must).

Note that the modal modifier construction is added to the inventory but not the network as it is not unique to a single word sense or lexical construction.

After learning the lexical, argument structure and word sense constructions from these 11 annotated sentences, the resulting construction inventory contains 19 constructions. Out of these 19 constructions, there are 5 lexical, 6 word sense and 8 argument structure constructions. The final categorial network is depicted in Figure 18. This network can then be used to comprehend and extract frames from English sentences.

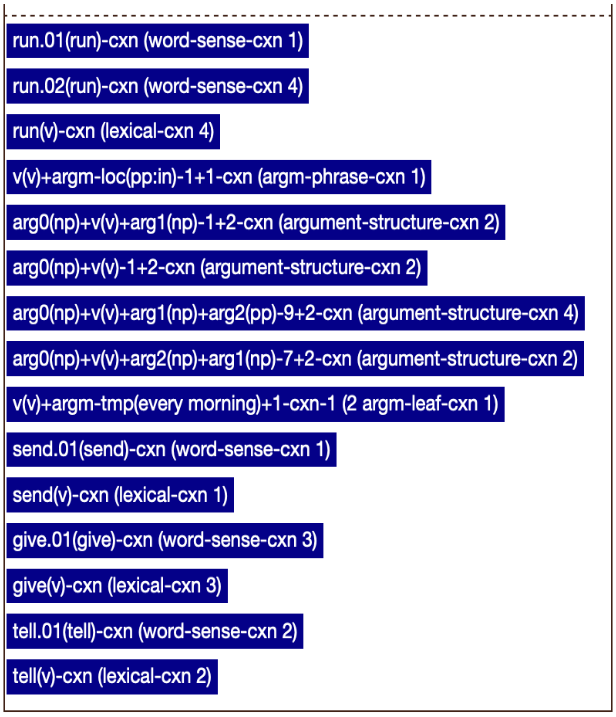

Figure 14: Construction inventory after learning the "run" sentences

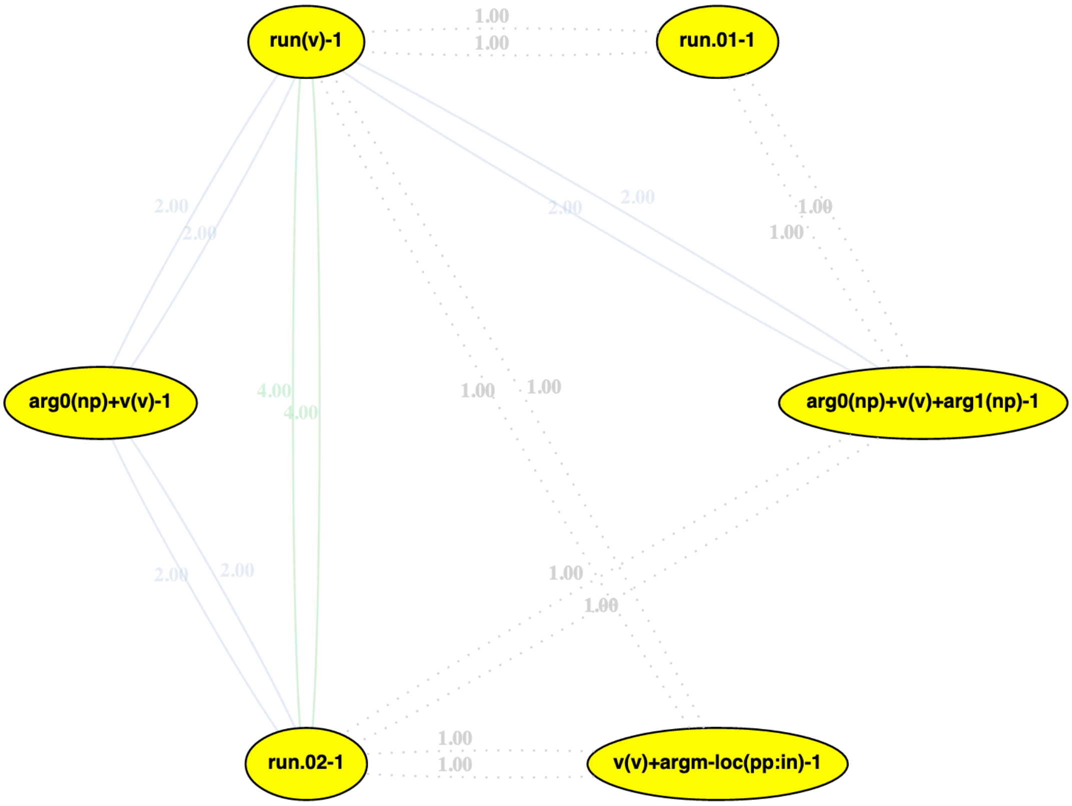

Figure 15: Categorial network after learning the "run" sentences

Figure 16: Construction inventory after learning "We must prepare for the exams"

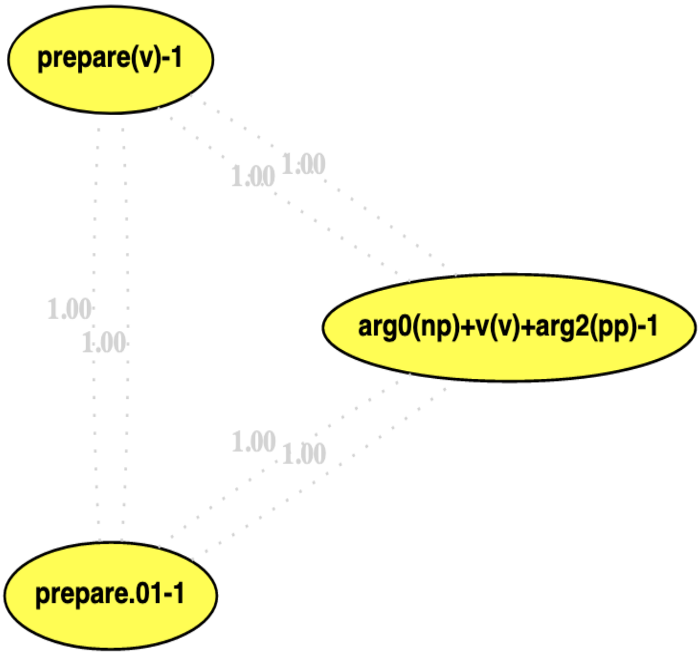

Figure 17: Categorial network after learning “We must prepare for the exams”

B Comprehending and Extracting Frames with a PropBank Grammar

This task is divided into two main steps: comprehension and frame extraction. After this general overview, more details are provided regarding the heuristics that guide the construction application.

The comprehension phase includes the following steps:

1. **De-rendering the Sentence**: The sentence is turned into an initial transient structure, a preliminary representation based on dependency and constituency structures. In a computational PropBank construction grammar, de-rendering refers to the conversion of an utterance into an initial transient structure. This structure is derived using Spacy’s dependency analysis and Benepar’s constituency analysis. Firstly, an utterance, either as a string or a CoNLL-formatted sentence, is processed. If the utterance is a string, a syntactic analysis can be performed if desired. Following this, the initial transient structure is created from the Benepar analysis. Each node in the analysis is turned into a ‘unit’ in the transient structure. This unit holds important properties of the node, such as its type, string, span, parent, and syntactic class. The initial transient structure serves as a foundation from which the application of constructions can begin. To illustrate this, the inital transient structure at the start of the comprehension process for sentence “Sarah gives Peter a new watch” Table 16 is shown in Figure 19. It is important to point out that the comprehension process operates on unannotated sentences. The annotations visible in Table 16 are illustrative.

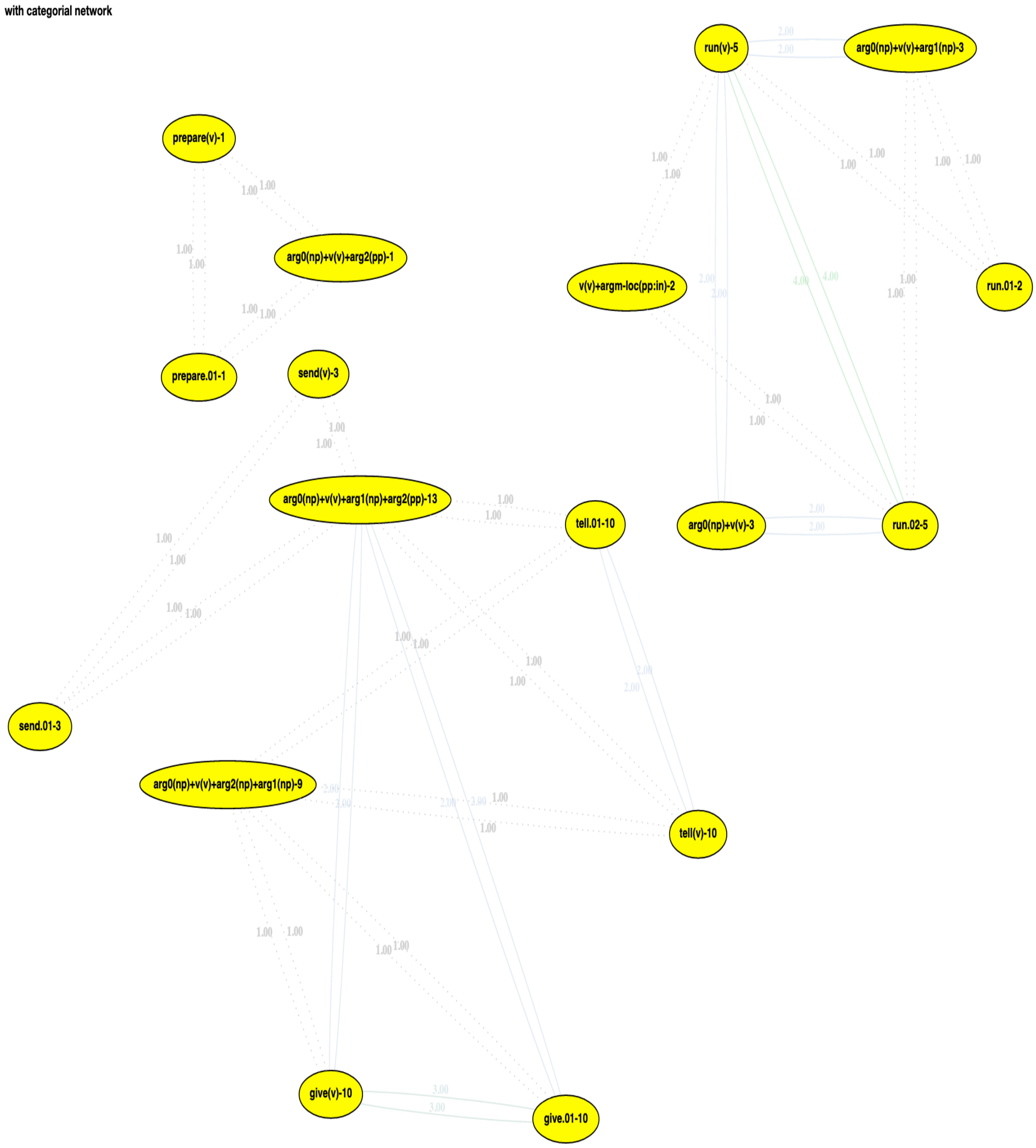

Figure 18: Full categorial network after learning the sentences from the custom corpus

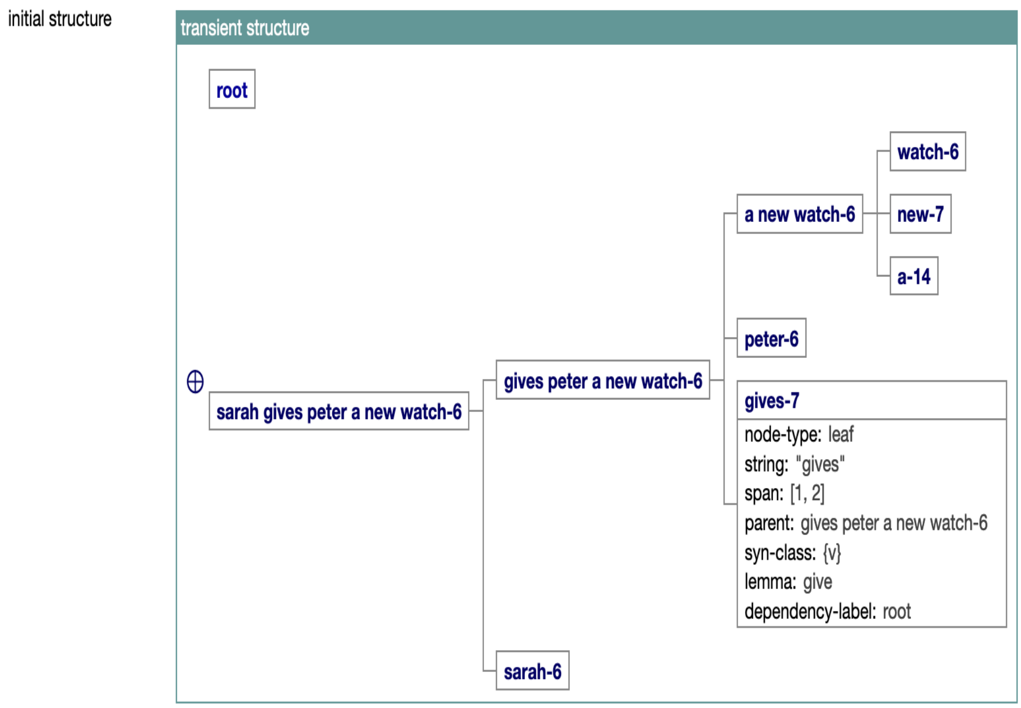

Figure 19: Initial transient structure after de-rendering. The "gives" unit is expanded to highlight which information is present before the application of constructions.

Table 16: PropBank annotations for "Sarah gives Peter a new watch"

String	**Sense**	**Lemma**	**Roles**
Sarah	-	-	ARG0
gives	give.01	give	V
Peter	-	-	ARG2
a new watch	-	-	ARG1

The initial transient structure for "Sarah gives Peter a new watch" is the result of the de-rendering process. It shows the information attached to the units before the construction application begins. The units contain mostly syntactic information, for example, node-type, parent and syntactic class. Note that before constructions are applied no information on the meaning is embedded.

2. **Application of Constructions**: Constructions from the inventory are applied to the initial transient structure, expanding it to capture more meaning. The selection of constructions to be applied is managed by the internal structure of the construction inventory and the construction supplier. Additionally, "heuristics", a strategy used to guide construction application, helps regulate this process. These heuristics assign scores to constructions during application, influencing which ones are applied but not altering the internal structure of the inventory. The specific heuristics used can lead to different constructions being applied. These heuristics will be discussed in more detail later. An example of the application process for sentence "Sarah gives Peter a new watch" is visible in Fig-

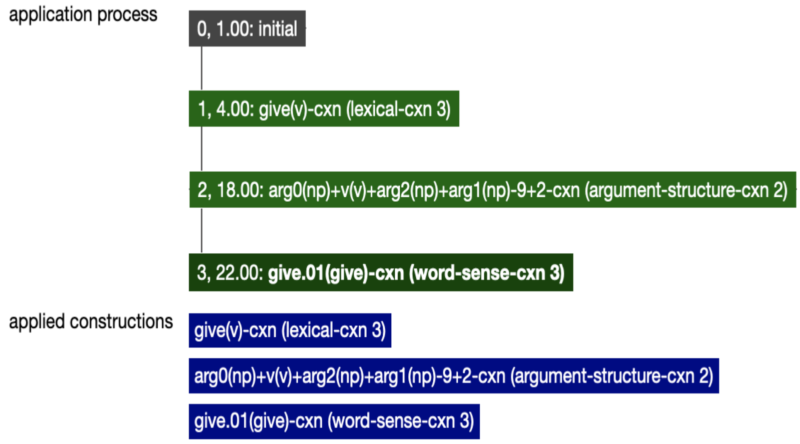

Figure 20: The construction application process during comprehension for "Sarah gives Peter a watch".

ure 20.

The application process for "Sarah gives Peter a new watch" included applying a lexical, argument structure and word sense construction. These constructions were used to expand the information embedded in the units of the transient structure.

3. **Goal Test**: The goal test is a method used within the comprehension process to determine whether a node has been fully and accurately expanded. It is implemented in a way that ensures that no more applicable constructions are present when a node has reached its complete expansion and that no constructions can be applied to its child nodes.
4. **Meaning Extraction**: The final transient structure is used to create a semantic network, which contains the sentence's semantic content. The final transient structure for sentence "Sarah gives Peter a watch" is shown in Figure 21 and the resulting meaning is shown in Figure 22.

After the application of constructions, a lot more information is embedded in the "gives" unit. Now, it not only includes syntactic information but also the meaning, whether it is a frame-evoking element or not and the frame name is present in the unit's body. The comprehended meaning shows the labeled semantic roles which match the roles shown in Table 16.

After this, the process moves to the frame extraction phase. Here, the final transient structure is used to extract frames, providing a structured representation of the situations or events described in the sentence. In the **Frame Extraction** process, an initial set of frames is pulled out from the final transient structure. This process involves examining each unit within the structure to determine if it has the capacity to evoke a frame. When such a unit is found a new frame is formed. This frame includes the name of the unit, the element that evokes the frame, and any associated frame elements:

- **Finding the Frame Name**: This step includes searching for a feature within the unit that signifies 'meaning', and within this feature, locating a label that specifies 'frame'.
- **Identifying the Frame-Evoking Element**: Here, the component of the unit that triggers or instigates the frame is singled out. This usually involves pinpointing

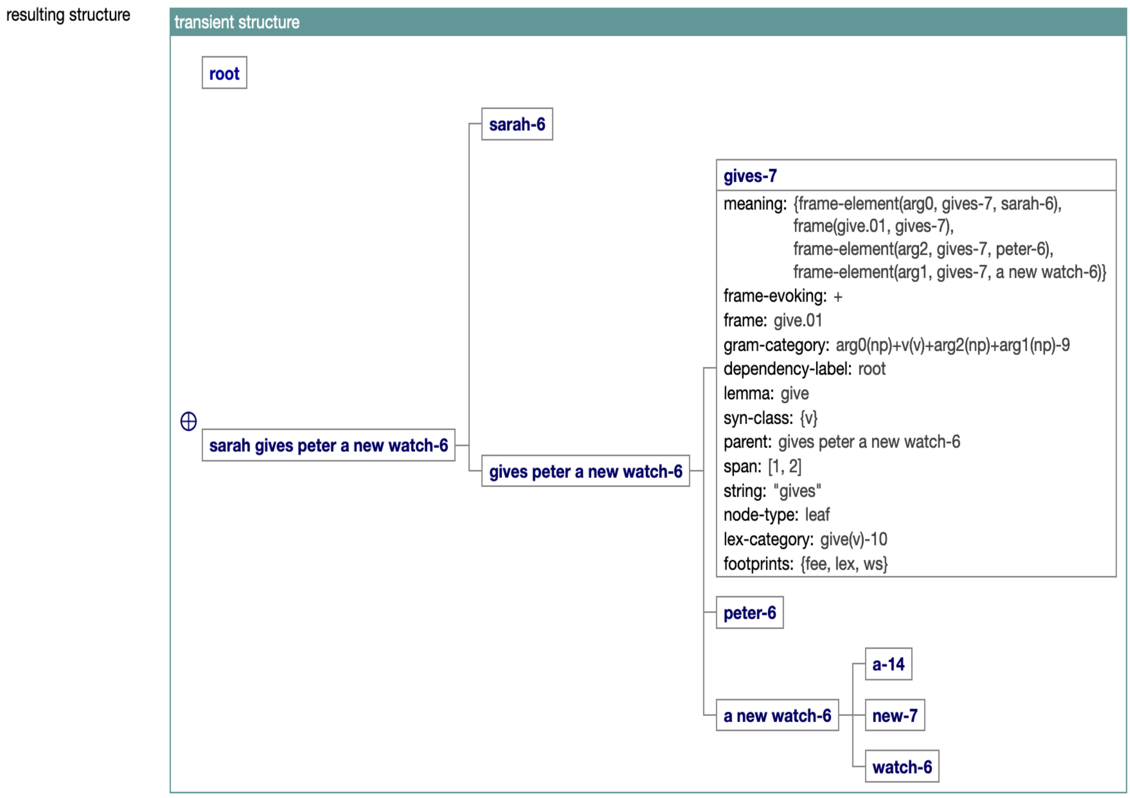

Figure 21: Final transient structure for "Sarah gives Peter a watch". The "gives" unit is expanded to highlight which information was added after the application of constructions.

Meaning:

(frame-element arg0 gives-7 sarah-6)

(frame give.01 gives-7)

(frame-element arg2 gives-7 peter-6)

(frame-element arg1 gives-7 a new watch-6)

Figure 22: Resulting meaning after comprehension for "Sarah gives Peter a watch".

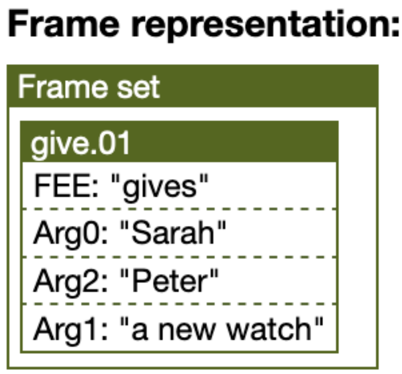

Figure 23: Resulting frameset after frame extraction for “Sarah gives Peter a watch”.

the features of the unit denoted as ‘span’ and ‘string’, which are correspondingly linked to the indices and actual string of the frame-evoking element.

- **Identifying Frame Elements**: This part of the process recognizes the elements associated with the frame within the transient structure. These elements depict the participants or entities involved in the action of the frame. The identification of frame elements is achieved by looking for labels signifying ‘frame-element’ in the feature of the unit representing ‘meaning’. For every ‘frame-element’ label found, the corresponding unit is identified within the transient structure and a new frame element instance is created. This instance includes the name, role, string, and indices of the frame element.

After all these steps, the frameset (a collection of frames) is created and returned. The frameset for the sentence “Sarah gives Peter a watch” is shown in Figure 23. The frameset for this sentence contains a single frame “give.01” which has “gives” as Frame Evoking Element, “Sarah” as Arg0/giver, “Peter” as Arg2/entity given to, and “a new watch” as Arg1/thing given.