A Scalable Pipeline for Estimating Verb Frame Frequencies Using Large Language Models

Adam M. Morgan
NYU Grossman School of Medicine
550 1st Ave, New York NY 10016
Adam.Morgan@NYULangone.org

Adeen Flinker
NYU Grossman School of Medicine
550 1st Ave, New York NY 10016
NYU Tandon School of Engineering
6 MetroTech Center, Brooklyn NY 11201
Adeen.Flinker@NYULangone.org

Abstract

We present an automated pipeline for estimating Verb Frame Frequencies (VFFs), the frequency with which a verb appears in particular syntactic frames. VFFs provide a powerful window into syntax in both human and machine language systems, but existing tools for calculating them are limited in scale, accuracy, or accessibility. We use large language models (LLMs) to generate a corpus of sentences containing 476 English verbs. Next, by instructing an LLM to behave like an expert linguist, we had it analyze the syntactic structure of the sentences in this corpus. This pipeline outperforms two widely used syntactic parsers across multiple evaluation datasets. Furthermore, it requires far fewer resources than manual parsing (the gold-standard), thereby enabling rapid, scalable VFF estimation. Using the LLM parser, we produce a new VFF database with broader verb coverage, finer-grained syntactic distinctions, and explicit estimates of the relative frequencies of structural alternates commonly studied in psycholinguistics. The pipeline is easily customizable and extensible to new verbs, syntactic frames, and even other languages. We present this work as a proof of concept for automated frame frequency estimation, and release all code and data to support future research.

1 Introduction

Word (or *lexical*) frequency is one of the most widely-used constructs in natural language research. In NLP, explicit use of word frequency estimates has driven major improvements in language models, increasing model speed by orders of magnitude (Mikolov et al., 2011, 2013) and bringing models closer to human-like performance benchmarks (Pennington et al., 2014; Gong et al., 2018). In research on human cognition, it is one of the strongest predictors of behavior (e.g., response times (Balota and Chumbley, 1984; Brysbaert et al., 2018), event-related potentials (Van Petten and Kutas, 1990)). Its use led to discoveries of phenomena like the frequency-by-regularity interaction (Seidenberg and McClelland, 1989), which paved the way to the development of the connectionist architecture underlying nearly all modern models of human cognition (Smolensky, 1988).

Where lexical frequency has proved invaluable for understanding word-level information, Verb Frame Frequencies (VFFs) – the frequency with which a verb takes particular sets of arguments like a direct or indirect object[1] – offer a powerful window into syntax, which remains far less understood than words. For instance, in behavioral research (e.g., Trueswell and Kim, 1998; Berkovitch and Dehaene, 2019), VFFs have been used to demonstrate that verb representations are inextricably linked to the structures they co-occur with, supporting "lexicalist" theories of syntax (MacDonald et al., 1994; Levin and Hovav, 1994; Pickering and Branigan, 1998; Ryskin et al., 2017), and in neuroscientific research (e.g., Meltzer-Asscher et al., 2015; Shetreet et al., 2007) they have been leveraged to map syntactic and semantic information in the brain.

However, relatively few studies have leveraged VFFs, and syntax remains far less understood than words. The main limiting factor is the limited availability of high-quality VFF estimates. That is, while there exists a vast number of tools and databases for calculating words' frequencies (e.g., Brysbaert and New, 2009; Baayen et al., 1996; Davies, 2008; Michel et al., 2011; Balota et al., 2007; Coltheart, 1981), there are very few resources for obtaining good estimates of VFFs. Some resources provide detailed inventories of verb-frame types and fine-grained semantic distinctions (e.g., FrameNet (Ruppenhofer et al., 2016), VerbNet (Schuler, 2005), VerbAtlas (Di Fabio et al., 2019)), but they do not quantify how frequently

[1] We define verb frames (also referred to as *argument structures*, *subcategorization frames*, or *complement structures*) as the *selected* arguments of the verb – i.e., excluding optional modifiers like adjuncts and non-selected adverbs/adjectives.

individual verbs appear in each frame, a critical distinction in settings where frequency modulates cognitive processing (e.g., syntactic priming, disambiguation, language acquisition). Previous studies using VFFs have largely run their own norming tasks to obtain estimates (e.g., Trueswell and Kim, 1998; Garnsey et al., 1997; Ryskin et al., 2017), requiring significant time and effort. And while many such studies have made the raw data publicly available, these datasets are often small, involving just the subset of verbs and frames relevant to the particular study. Despite major advances in large-scale lexical and semantic resources, no existing tool provides scalable, empirically-grounded estimates of verb frame frequencies.

The primary obstacle to obtaining high-quality VFF estimates is the difficulty of *parsing*, or identifying the underlying syntactic structure in a string of words. Manual parsing, where trained linguists analyze and annotate a corpus (creating a "treebank") is the gold standard, but it is prohibitively time- and resource- intensive. A number of automated approaches exist (e.g., Petrov and Klein, 2007; Qi et al., 2020; Kitaev and Klein, 2018), but these "parsers" still require manual checking (Taylor et al., 2003) and suffer from systematic biases, including high error rates for certain complex structures (Rimell et al., 2009; Choi et al., 2015) and in different language registers (e.g., informal speech; Yang et al. 2015). Consequently, while automated parsing provides valuable scalability and efficiency, the gold standard for estimating VFFs remains manual annotation.

In the most comprehensive set of manual VFFs to date, Gahl et al. (2004) had four trained linguists parse 200 sentences for each of 281 American English verbs. They focused in particular on verbs implicated in a widely-studied source of processing difficulty: the Noun Phrase (NP)/Sentential Complement (SC) ambiguity, where a verb like *accept*, which can take a NP or SC complement, is immediately followed by a NP like "the money," which is temporarily ambiguous between an NP complement or the subject of an SC:

(1) *The NP/SC Ambiguity:*

a. accept [the money]$_{\text{NP}}$ *NP*

b. accept [the money is gone]$_{\text{SC}}$ *SC*

They then compared their results to those of 10 earlier studies, demonstrating overall high agreement. These frequencies have proven of immense value, having been used in dozens of subsequent behavioral and neuroscientific investigations of language processing (e.g., Staub et al., 2006; Linzen and Jaeger, 2016; Vuong and Martin, 2015).

However, there are some notable gaps in the Gahl et al. (2004) dataset. For one, Gahl et al. collapsed across intransitive and Prepositional Phrase (PP) frames (e.g., *"look at me"* or *"look for the remote"*), reducing the accuracy of their dataset for two of the most common verb frames. Perhaps even more importantly, they did not include verbs or frames implicated in one of the most commonly-studied phenomena in psycholinguistics: *structural alternations*, where the production system can express the same meaning with different verb frames. For instance, dative verbs like *loan* can take a "Direct Object" (DO; Ex. 2a) or "Prepositional Object" (PO; Ex. 2b) frame, and locative verbs like *load* can appear with an "On" (Ex. 3a) or "With" frame (Ex. 3b). Consequently, their dataset lacks many common verbs (e.g., *give*, *put*, and *show*) and frames that are among the most important in psycholinguistic research.

(2) *The Dative Alternation:*

a. loan [the kid]$_{\text{NP}}$ [a book]$_{\text{NP}}$ *DO*

b. loan [a book]$_{\text{NP}}$ [to the kid]$_{\text{PP}}$ *PO*

(3) *The Locative Alternation:*

a. load [hay]$_{\text{NP}}$ [onto the truck]$_{\text{PP}}$ *"on"*

b. load [the truck]$_{\text{NP}}$ [with hay]$_{\text{PP}}$ *"with"*

Critical to understanding how producers choose between competing structures is knowing the relative frequency of the competing alternates. However, few resources exist for these alternating frames. Some studies have attempted to circumvent the difficulty of parsing large datasets. For instance, Hawkins et al. (2020) used LLM surprisal values to assess the relative preference of English dative verbs for the Direct Object (DO) and Prepositional Object (PO) frames (2). To evaluate performance, rather than running a production study or parsing a corpus, Hawkins et al. (2020) asked human participants to use a slider to indicate their relative preference for a DO sentence vs. its corresponding PO formulation. In not requiring transcription and parsing, this approach enables the rapid collection and analysis of a huge amount of data. Their results showed strong correlations between the human preference ratings and relative surprisals for

a number of the language models, explaining 73% of the variance with GPT2-large. However, questions remain about the interpretation of relative preference ratings, including their degree of psychometric validity and how exactly they relate to frequency (cf. Myers, 2017; White and Rawlins, 2020).

In sum, there are three main limitations in extant tools for estimating VFFs. First is the trade-off between high quality parses and scalability: manually annotated datasets are the gold-standard for calculating VFFs, but being extremely time- and labor-intensive, these cannot readily be adapted or extended to, e.g., additional verbs, frames, or other languages. Second, existing datasets have important gaps, either in the granularity of their syntactic distinctions (as in Gahl et al.'s (2004) collapsing of intransitive and PP frames) or in the breadth of their coverage (e.g., the absence of dative and locative verbs and frames). Third, existing studies have used vastly different methodological approaches (e.g., manual annotation, relative acceptability rating, LLM surprisals), making it difficult to evaluate findings across studies or combine results to build more comprehensive datasets.

Here we aim to overcome these limitations by leveraging recent advances in artificial intelligence to create an automated pipeline for calculating VFFs that is fast, accurate, customizable, and scalable. We began by compiling a more comprehensive list of verbs, including all 281 verbs in Gahl et al. as well as 195 additional verbs implicated in the dative and locative alternations. We then automated the creation of a mini-corpus, using an LLM to repeatedly generate sentences given the entire set of 476 verbs. By instructing the LLM to behave like an expert linguist, we obtained parses for each sentence, converting our corpus into a treebank. To evaluate how well this pipeline performed relative to existing tools, we parsed the same sentences with two commonly-used constituency parsers: the Berkeley Neural Parser (*benepar*; Kitaev and Klein 2018; Kitaev et al. 2019) and Stanford CoreNLP (Manning et al., 2014). We then compare the results from the LLM and the two existing parsers to previously published datasets and show that in nearly every case, the LLM significantly outperforms the Berkeley and Stanford Parsers. Alongside this manuscript, we make available all scripts used in this pipeline, as well as the raw, preprocessed, and categorized parses. By making VFF estimation fast and scalable, our approach enables future researchers to choose their own level of granularity among syntactic distinctions, and to readily scale up to other verbs, frames, and even other languages.

2 Methods

2.1 Data Sources

Verb selection. We selected 476 English verbs from established lexical databases frequently used in linguistic and psycholinguistic research. This included all 281 verbs used in Gahl et al.'s (2004) gold-standard dataset of American English verb frame frequencies, as well as verbs included in previous studies of the dative alternation (Hawkins et al., 2020; Theijssen et al., 2009), locative alternation, and NP-X ambiguity (Trueswell et al., 1993; Garnsey et al., 1997), as well as those associated with these alternations in Levin, 1993. These sources were chosen for their broad acceptance and foundational role in verb argument structure and syntactic processing research.

Context Generation. We used Open AI's GPT-o1 to generate 1,000 brief task contexts (see Appendix A for the complete prompt). The output included contexts describing various locations, situations, and times, including, e.g., "on the tennis court" and "down to the wire."

Sentence Generation via LLM. Sentences using each verb were created using Open AI's GPT-4o-mini (OpenAI, 2024a). The system message instructed the model: "You are a random sentence generator, tasked with generating natural-sounding sentences like those you might find in a conversation, movie, or newspaper. The random sentence generator should provide sentences with varied meanings, tenses, and syntactic structures, simulating random draws from sentences anywhere on the internet – including in movies, newspapers, conversations, forums, etc. to produce naturalistic, plausible sentences using a specific verb and context." It further specified, "Responses do not have to include words from the context, but please ensure they are thematically related," and listed four example input-output pairs (see Appendix B for the full system message). A total of 100 batch prompts were sent via the API, each of which included the full list of 476 verbs along with a single context, generating 100 sentences for each verb.

Risk statement. This project poses minimal risk. However, because it generates and parses sen-

tences, biases in the LLM's training data likely leads to over- or underrepresentation of dialect-specific verb biases. By releasing all code and data, we aim to promote transparency. Future work should assess and mitigate such biases.

2.2 Parsing

Sentence Preprocessing. Prior to parsing, sentence strings were split by commas, which typically indicated root-level clause boundaries. The resulting substrings that did not include the main verb were discarded.

The Berkeley Neural Parser. The clause containing the target verb was parsed using the Berkeley Neural Parser (benepar; Kitaev and Klein, 2018), implemented via the spaCy interface with the benepar_en3 model (see Script 3b in the repository). Benepar assigns Penn Treebank–style constituency structures to text using a neural chart parser trained on annotated corpora. It maps words to hierarchical phrase structures annotated with clause- and phrase-level labels from the Penn Treebank tagset (Taylor et al., 2003).

The Stanford CoreNLP Constituency Parser. The clause with the target verb was also passed to the Stanford CoreNLP constituency parser (version 4.5.7) using the NLTK interface (version 4.2.0; Bird, 2006) (see Script 1 in the repository). Like benepar, the Stanford Parser maps words to hierarchical structures, with nodes labeled using the Penn Treebank tags.

The LLM (GPT-4o) Parser. Argument structures were also annotated using GPT-4o (specifically: gpt-4o-2024-05-13; OpenAI, 2024a). Initial attempts at feeding all sentences for each of the 100 sentence generation batches to the model in a single batch resulted in the model omitting a large proportion of the input sentences in its response. Trial and error revealed that the model was more likely to return a parse for each input sentence when the number of sentences was 100 or fewer, so we split the prompts into sets of 100 or fewer sentences.

The system message (which appears verbatim in Appendix C) instructed GPT-4o that it was an expert linguist who would be given multiple sentence-verb pairs. For each entry, its task was to isolate the clause containing the given verb, remove everything outside of the Verb Phrase (VP) as well as any optional modifiers like "time, manner, or location expressions that are not required or licensed by the verb," and to return just the arguments that the verb subcategorizes for. We provided several examples, which included a diversity of argument structures as well as situations where optional arguments (e.g., adjectives) should be ignored. The model was instructed to tag each argument using the standard Penn Treebank tags to facilitate comparison to the output of the Stanford and Berkeley Parsers. We explicitly instructed the model not to infer missing arguments to prevent the model from annotating a sentence like "I ate" as transitive (i.e., having an object like "dinner"), given that, even though there is an implicit semantic object, syntactically this is an intransitive use of the verb *eat*. The full system message is provided in Appendix C.

2.3 Data Cleaning

Prior to categorizing argument structures, we cleaned the responses from both the Stanford, Berkeley, and GPT-4o parsers (Script 4). To avoid potential hallucination of arguments in the LLM parses, we explicitly verified that every argument the LLM identified was present in the original sentence. We excluded sentences where the target verb was incorrectly used as a noun (e.g., *party*; "I never thought a party could turn so somber."). Prepositional phrases were labeled by their particular preposition to distinguish between ordinary prepositional arguments (e.g., "cook [carrots]$_{NP}$ [on the stove]$_{PP}$") and those that are implicated in the alternations of interest (e.g., *for*-phrases in benefactive constructions like "cook [carrots]$_{NP}$ [for the children]$_{PP}$" – which could alternatively appear in the Double Object construction: "cook [the children]$_{NP}$ [carrots]$_{NP}$). In all, these steps removed 3,087 sentences from the dataset (4.779% of the original 64,589).

Finally, we excluded 75 sentences where the target verb appeared in a passive construction. This decision was motivated by two concerns. First, both parsers showed inconsistent behavior in annotating passives, leading to unreliability in frame assignment. Second, there is no clear consensus on how passives should be categorized: while syntactically intransitive, many linguistic theories (e.g., Chomsky, 2014a,b) argue that they are derived from underlying transitive structures. Given these inconsistencies and theoretical ambiguities, and because passives represented a small proportion of the data, we excluded them to improve comparability and reliability of frame estimates.

For verbs implicated in the Dative or Locative

alternations, we separately estimated their rates of each alternating structure. This involved certain additional criteria. As the PO dative alternation (Ex 2b) only involves PPs headed by *to* or *for*, we excluded PPs headed by other prepositions. Similarly, the locative alternation only involves a subset of prepositions for each verb (e.g., *with*, *on(to)*, *in(to)*, *over*, etc.). We excluded NP-PP frames where the PP was headed by a preposition that the specific verb does not take in locative constructions.

2.4 Argument Structure Extraction and Frame Categorization

To calculate VFFs, we counted the occurrences of each of the unique argument structures across the cleaned dataset. To simplify matters, we established a minimum threshold for the number of times each argument structure had to appear in the dataset to be counted as a unique frame (specifically, the 75th percentile of the frame counts, or 65 instances). We binned all frames that did not meet this threshold into an "other" category.

2.5 Evaluation and Comparison to Existing Norms

Our automated pipeline's output was validated against the gold-standard Verb Frame Frequencies reported in Gahl et al. (2004). Gahl et al.'s dataset consisted of manually parsed sentences (200 sentences for each of 281 verbs), providing a reliable benchmark for assessing our method's accuracy. Comparative analyses included quantitative evaluations of frame frequency distributions and qualitative examinations of discrepancies attributable to methodological differences.

Our method diverged from Gahl et al.'s (2004) in two important ways. First, due to the importance of structural alternations in the psycholinguistic literature, we added a number of verbs to our dataset implicated in either the locative alternation (verbs like *stuff* and *spray*) or the dative alternation (including "transfer" verbs like *give* and *hand* and "benefactive" verbs like *bake* and *buy*). Second, whereas Gahl et al. treated verbs which select for Prepositional Phrases (PPs; e.g., "look [for my keys]$_{PP}$") as intransitive, here we treat these as their own subcategorization frame, distinct from intransitive and NP frames. For the purposes of evaluating our pipeline's output, we relabeled all PP frames as intransitive prior to comparison to Gahl et al.'s data.

3 Results

All data, scripts, and results are available online at https://www.doi.org/10.17605/OSF.IO/FRQBE.

3.1 Unique Subcategorization Frames

After cleaning the data (Section 2.3) and binning low-frequency frames into an "other" category (Section 2.4), we were left with 81 unique frames. Figure 1 shows the number of instances of the 5 most common frames. Sample sentences for each of these appear in Appendix D. Notably, two of these frames involve PP arguments, which Gahl et al. (2004) excluded from their frame counts. By contrast, our approach treats selected PPs as syntactically licensed arguments, resulting in a more fine-grained and psycholinguistically relevant set of verb frame categories (see Section 3.5).

Figure 1: Counts of the five most frequent verb frames (out of 61,427 sentences), by parser.

3.2 Evaluating the Parsers

Despite being applied to the same set of sentences, GPT-4o produced the same parse as the the Berkeley Parser for only 45% of sentences, and only 37% for the Stanford Parser. To evaluate accuracy, we compared results from all three parsers to the manually parsed (i.e., gold-standard) Gahl et al. (2004) dataset. (The Berkeley Neural Parser being state-of-the-art, we report these comparisons in the main text and provide Stanford results in the supplement for comparison.) Figure 2 shows the relationship between the proportion of NP (i.e., transitive) tags between each verb in Gahl et al. and the GPT-4o (left) and Berkeley (right) parsers. We found significant relationships for both GPT-4o (linear regression, $t(245) = 13.225$, $p < .001$)

and Berkeley (linear regression, $t(245) = 7.502$, $p < .001$), although the GPT-4o VFFs accounted for over twice as much variance ($r^2 = .417$) as the Berkeley VFFs ($r^2 = .187$).

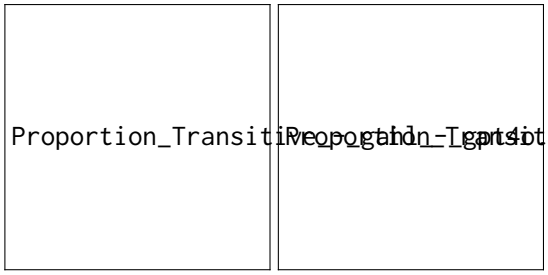

Figure 2: Evaluating the GPT-4o and Berkeley Parsers' VFF estimates. We compared estimates to Gahl et al.'s (2004) gold-standard dataset. Here we show these comparisons for the NP frame (see Appendix E for others). Results showed significant correlations for both parsers, but GPT-4o (left) produced a better fit ($r^2 = .406$) than the Berkeley Neural Parser (right; $r^2 = .187$).

We repeated this analysis for each of the seven modal frames in Gahl et al.'s dataset (see Appendix E). This resulted in a total of 14 "univariate" models: seven structures $\times$ two parsers. Our results showed that both parsers performed above chance: the GPT-4o VFFs significantly predicted Gahl et al.'s (2004) for all seven frames, and the Berkeley VFFs were significant for six frames (see Table 2 in Appendix E). However, relative to the Berkeley estimates, GPT-4o estimates accounted for more variance in the Gahl et al. (2004) data for every structure we tested – on average, 1.454 times more, and 2.010 times more than the Stanford estimates.

To statistically compare how well the GPT-4o and Berkeley Parsers fit Gahl et al.'s (2004) data, we performed a series of model comparisons. We started by re-fitting the linear models using Bayesian regression (Bürkner, 2017) and, for each verb frame, calculated the Bayes Factor (BF), or how many times more evidence there was for the GPT-4o VFFs than the Berkeley VFFs. For all verb frames, there was "strong" or "decisive" evidence (log_{10}(BF) > 1 and > 2, respectively; Jeffreys 1998) in favor of the GPT-4o VFFs over the Berkeley VFFs. Taken together, these results suggest that GPT-4o, when prompted appropriately, can outperform the most widely used automated parsers.

3.3 Human Validation

To evaluate parser accuracy against human annotations, an expert linguist manually annotated 300 sentences – 100 for each of three representative verbs: *realize*, *believe*, and *loan*. We then compared these manual annotations to the frame predictions made by each of the three automated parsers. GPT-4o achieved the highest match rate with manual annotations for all three verbs (Fig. 3), agreeing on 79% of sentences overall, compared to 69% for Berkeley and 59% for Stanford. Critically, GPT-4o agreed with the human annotations on 74% of sentences for the verb *loan*, which is not present in Gahl et al.'s dataset, demonstrating generalizability to a dative verb (Ex. 2) commonly used in psycholinguistic experiments.

Structure	Winning Parser	log_{10}(BF)	Degree of Evidence
Transitive	GPT-4o	17.490	Decisive
Intransitive	GPT-4o	29.521	Decisive
Sentence	GPT-4o	1.328	Strong
Particle	GPT-4o	3.301	Decisive
NP-Sentence	GPT-4o	7.351	Decisive
Non-finite	GPT-4o	3.454	Decisive
Particle-NP	GPT-4o	5.716	Decisive

Table 1: Model comparisons. For seven structures, we compared models predicting Gahl et al.'s (2004) gold-standard VFFs using the GPT-4o or Berkeley VFFs. The resulting Bayes Factors (BFs) indicate how much more evidence there was for the GPT-4o VFFs than the Berkeley VFFs (positive values in log-space) or vice versa (negative). BFs were interpreted according to the Jeffreys (1998) scale. For six of the seven structures, we found "decisive" evidence that the GPT-4o estimates fit Gahl et al.'s VFFs better than the Berkeley estimates. For the sentence frame, evidence was merely "strong."

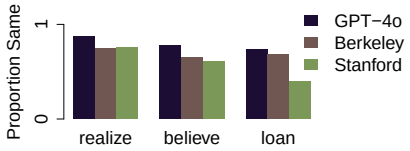

Figure 3: Human validation: The proportion of sentences for which each parser agreed with human annotations for three representative verbs (100 sentences per verb). GPT-4o consistently outperformed the other two.

3.4 The NP/SC Ambiguity

Given the importance of the NP/SC ambiguity in psycholinguistic research, we sought to validate our estimates of verb biases for these two frames. We started by excluding all responses that were not categorized as either an NP or SC completion for each

parser, and then calculating the log-odds of NP over SC completions for each of ~ 45 verbs used in two prior studies that published norming data. Figure 4 shows that both the GPT-4o- and Berkeley-based estimates significantly predicted the results from Trueswell et al. (1993) (GPT-4o: $r^2 = .468$, $p < .001$; Berkeley: $r^2 = .321$, $p < .001$) and Garnsey et al. (1997) (GPT-4o: $r^2 = .390$, $p < .001$; Berkeley: $r^2 = .375$, $p < .001$). Bayesian model comparison revealed "decisive" evidence in favor of the GPT-4o estimates when modeling Trueswell et al.'s data (log_{10}(BF)$= 2.386$; Jeffreys 1998), but no noteworthy evidence in favor of either parser for modeling Garnsey et al.'s data (log_{10}(BF)$= .290$). Overall, while both parsers capture verbs' NP/SC biases to some extent, GPT-4o's estimates more closely align with those obtained from these human norming studies, reinforcing its potential as a tool for accurately estimating verb frame frequencies.

Figure 4: Evaluating GPT-4o (left column) and Berkeley Neural Parser (right) estimates of NP bias for NP/SC verbs. We compared our estimates to those from two prior studies: Trueswell et al. (1993; top row) and Garnsey et al. (1997; bottom). Both parsers significantly predicted the previous results, though the GPT-4o model accounted for more of the variance in both datasets.

3.5 Improving Estimates of Intransitivity

One shortcoming of Gahl et al.'s (2004) estimates is that they collapsed intransitive and PP frames. Beyond reducing syntactic granularity, this may have led to systematic errors in their estimates of intransitivity. To assess the impact on their results, we compared our intransitivity estimates to theirs in two ways: first, counting PP frames as intransitive (Fig. 5, top row), and next counting only verbs with no arguments as intransitive (bottom). As expected, excluding PPs from intransitivity counts produced worse fits for both parsers, reducing r^2 from .611 to .517 for the GPT-4o parser (left column) and from .339 to .151 for the Berkeley Neural Parser (right). These results suggest that separating PP frames improves the accuracy of intransitivity estimates, and that our method may provide more faithful estimates of intransitivity rates.

Figure 5: Evaluating the effect of including PP frames in intransitivity counts. For both the GPT-4o (left) and Berkeley (right) parsers, including PPs (as Gahl et al. (2004) did) produced a better fit (top) to Gahl et al.'s estimates than excluding them (bottom), meaning that Gahl et al. lost meaningful information by collapsing these categories.

3.6 The Dative Alternation

Another primary goal of this work was to provide high-quality VFF estimates for commonly studied verb frames that were not included in Gahl et al. (2004) – in particular, datives and locatives, which feature prominently in psycholinguistic research. We therefore added a number of verbs not present in the Gahl et al. dataset. We focused on two ditransitive frames that were not tracked in Gahl et al.: NP-NP (as in the Dative Direct Object frame; Ex. 2a) and NP-PP (as in the Prepositional Object frame; Ex. 2b).

To estimate verb-specific preferences for the two dative frames, we calculated the log-odds of DO over PO rates for the GPT-4o and Berkeley parses. Our results showed a wide range of preferences,

with some verbs strongly preferring the DO frame (e.g., *teach* as in "teach the dog a trick") and others preferring PO completions (*ship* as in "ship the crate to its owner").

To evaluate the accuracy of these estimates, we compared them to the relative preference ratings reported in Hawkins et al. (2020) for the 150 verbs our studies shared in common. Many verbs had no DO completions at all (the row of dots at the bottom of both panels in Fig. 6), consistent with Hawkins et al.'s inclusion of non-alternating PO verbs. As before, we performed a Bayesian model comparison, which revealed "decisive" evidence (log_{10}(BF)$= 8.700$; Jeffreys 1998) in favor of GPT-4o over Berkeley.

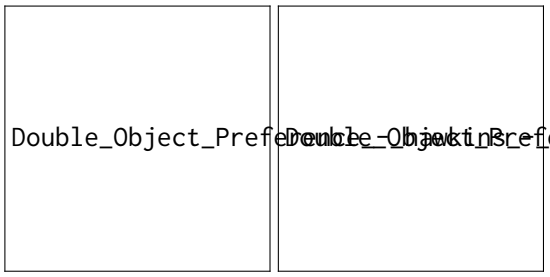

Figure 6: Evaluating the GPT-4o (left) and Berkeley (right) Parsers' estimates of DO/PD bias for dative verbs. We compared estimates to Hawkins et al.'s (2020) relative preference ratings. Many of the verbs they included are non-alternating (e.g., *whisper*), and for the majority of these neither GPT-4o nor the Berkeley parser identified any DO completions. To avoid infinities, we set proportions of 0 (no instances) or 1 to .001 and .999, respectively, capping log-odds values at ± 6.907 (the row of black dots at the bottom of each plot). Both parsers significantly predicted preference ratings, but GPT-4o accounted for more variance ($r^2 = .215$) than the Berkeley Neural Parser ($r^2 = .067$).

4 Discussion

Verb Frame Frequencies (VFFs) have proven central to understanding the cognitive and neural underpinnings of syntax, yet they remain underutilized—especially when compared to lexical frequencies. In machine language models, where they are nearly entirely unexplored, their explicit modeling stands to drastically improve model efficiency and performance as has been demonstrated for lexical frequency (Mikolov et al., 2011, 2013; Pennington et al., 2014; Gong et al., 2018). The primary obstacle to such research is the difficulty of deriving high-quality estimates of VFFs. Manual annotation remains the gold standard, but is prohibitively time- and labor-intensive. Automated parsers offer a scalable alternative, but often exhibit systematic biases, particularly for rare or structurally complex constructions (Rimell et al., 2009; Choi et al., 2015; Yang et al., 2015). What such datasets exist are limited in various ways: syntactic granularity (Gahl et al., 2004), interpretability of the metrics (Hawkins et al., 2020), or tailored to particular theoretical questions (Trueswell et al., 1993; Garnsey et al., 1997), curbing the degree to which they can be extended for more general use.

In this work, we introduced a fully automated pipeline for estimating VFFs, leveraging recent advances in artificial intelligence for both sentence generation and syntactic parsing. By prompting an LLM to generate diverse, contextually grounded sentences using a target verb, and then instructing it to parse those sentences using linguistic conventions, we created a syntactically annotated corpus of over 45,000 sentences spanning 476 English verbs. This dual-use of the LLM represents a novel contribution, enabling scalable VFF estimation without any manual intervention.

We benchmarked this approach against three major sources of VFF norms: Gahl et al. (2004) for broad coverage of argument types; Trueswell et al. (1993) and Garnsey et al. (1997) for NP/SC ambiguities; and Hawkins et al. (2020) for datives. The LLM parser significantly predicted human data in all three cases, and in nearly every comparison outperformed the Berkeley Neural Parser (and the Stanford Parser; see Supplementary Section F). These results validate both steps of the pipeline: the use of LLMs to simulate naturalistic syntax in generation, and their ability to produce linguistically coherent parses. The pipeline's ability to recover known alternation patterns across diverse verbs highlights its utility for psycho- and neurolinguistic research, where fine-grained verb frame frequencies are critical but difficult to obtain at scale.

More broadly, this work lays the foundation for future research using VFFs in both cognitive science and NLP. The resulting database includes more verbs, finer-grained distinctions, and broader coverage of structural alternations than any existing dataset. It can be readily extended to new verbs, languages, or syntactic phenomena. As tools like LLMs continue to improve, this pipeline could enable rapid, domain-specific estimation of syntactic preferences—supporting applications from psycholinguistic modeling to improved syntactic

generalization in neural architectures.

Limitations

This study constitutes a first attempt at building a verb-frame frequency database using large language models (LLMs) for both corpus generation and syntactic parsing. By automating both steps, the pipeline offers a scalable alternative to the previously labor-intensive processes of dataset creation, curation, and manual parsing. However, because the same model is used for generation and parsing, it is difficult to disentangle whether any observed limitations in accuracy stem from one stage or the other. Future work might separate these steps – e.g., by parsing human-written corpora or applying different models at each stage—to better isolate their contributions.

Our approach also involves a number of assumptions that merit further evaluation. To automate corpus creation for a targeted set of verbs, we used an LLM to generate sentences given those verbs. This assumes that the syntactic distributions produced by the model approximate those found in natural language environments. While this assumption is plausible given the size and breadth of LLM training corpora, the degree of variability in LLM outputs is modulated by hyperparameters that we did not manipulate (e.g., temperature, top_p, top_k). Our results showed strong correlations with existing corpora, but future work could improve performance by tuning these parameters to better match naturalistic distributions.

Similarly, we assumed that the model can provide reliable syntactic parses when instructed to behave like a linguist. While the LLM-generated parses aligned well with gold-standard data, this does not preclude the possibility of systematic biases in how the model analyzes sentence structure. Identifying such biases would be an important step toward improving accuracy further.

A related limitation is that the pipeline includes many degrees of freedom, any of which could impact the results. This leaves open many avenues for future research, for instance, evaluating the impacts of model parameters like temperature, nucleus sampling, token sampling, max tokens, and the frequency penalty; different model architectures (e.g., decoder-only transformers like GPT-3 or LLaMa vs. encoder-decoder models like T5 or BART); and input-level decisions (e.g., verb form, context inclusion, prompt phrasing, or the number of examples per request). Here we focused on introducing the method and demonstrating its potential for achieving high quality results with minimal manual effort. Future work should systematically explore these design variables.

An anonymous reviewer points out that, since both sentence generation and frame extraction are performed using models from the GPT family, it is possible that syntactic patterns in the generated data may align closely with the model's own parsing expectations, potentially inflating parsing success. To reduce potential circularity, we used different models for each stage: GPT-4o-mini for generation and GPT-4o for parsing. While both belong to the same family, they differ in architecture, capacity, and likely training exposure, reducing the chance that parsing simply reflects generation structure. Still, it is likely there is some degree of representational overlap. Future work will test this directly using models from entirely different families (e.g., LLaMA 3) for parsing.

One further limitation with respect to Gahl et al. (2004) is that we did not include passives. It remains unclear exactly how to determine which syntactic frame a passive verb belongs to: for instance, is "the melon was eaten" transitive, since the canonical object of "eat" is "melon"? Or is "eat" intransitive since "melon" is the syntactic subject here? And what is the relationship between each of these possibilities and the neural and model implementation of VFFs? In response to these unknowns, previous work like Gahl et al. has separately estimated VFFs for each possibility. In our case, both parsers showed inconsistent treatment of passives, further complicating analysis. For simplicity and clarity, we excluded them from the present study.

Finally, our parser comparison focused on a single baseline: the Stanford CoreNLP constituency parser. However, many widely used parsers exist, including spaCy (Honnibal et al., 2020), Stanza (Qi et al., 2020), AllenNLP, and Berkeley's neural parser (Kitaev and Klein, 2018). Each offers distinct strengths and may outperform the Stanford parser on certain constructions. Future research should benchmark LLM-based parsing against these alternatives, as it may be the case that even higher accuracy can be achieved from hybrid pipelines: for instance, using an LLM to generate a corpus given verbs/criteria of interest, and then using an existing automated parser to parse the output.

Ethics Statement

We take seriously ethical considerations in research and computing. The present work strictly abides by the ACL Ethics Policy.

Acknowledgements

We thank Andrew Chang and Jean Pouget-Abadie for invaluable guidance on this project.

References

R Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1996. The CELEX lexical database.

David A Balota and James I Chumbley. 1984. Are lexical decisions a good measure of lexical access? the role of word frequency in the neglected decision stage. *Journal of Experimental Psychology: Human perception and performance*, 10(3):340.

David A Balota, Melvin J Yap, Keith A Hutchison, Michael J Cortese, Brett Kessler, Bjorn Loftis, James H Neely, Douglas L Nelson, Greg B Simpson, and Rebecca Treiman. 2007. The english lexicon project. *Behavior research methods*, 39:445–459.

Lucie Berkovitch and Stanislas Dehaene. 2019. Subliminal syntactic priming. *Cognitive psychology*, 109:26–46.

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 interactive presentation sessions*, pages 69–72.

Marc Brysbaert, Paweł Mandera, and Emmanuel Keuleers. 2018. The word frequency effect in word processing: An updated review. *Current directions in psychological science*, 27(1):45–50.

Marc Brysbaert and Boris New. 2009. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior research methods*, 41(4):977–990.

Paul-Christian Bürkner. 2017. brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, 80:1–28.

Jinho D Choi, Joel Tetreault, and Amanda Stent. 2015. It depends: Dependency parser comparison using a web-based evaluation tool. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 387–396.

Max Coltheart. 1981. The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, 33(4):497–505.

Mark Davies. 2008. The corpus of contemporary american english (COCA): 560 million words, 1990-present.

Andrea Di Fabio, 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)*, pages 627–637.

Susanne Gahl, Dan Jurafsky, and Douglas Roland. 2004. Verb subcategorization frequencies: American english corpus data, methodological studies, and cross-corpus comparisons. *Behavior Research Methods, Instruments, & Computers*, 36(3):432–443.

Susan M Garnsey, Neal J Pearlmutter, Elizabeth Myers, and Melanie A Lotocky. 1997. The contributions of verb bias and plausibility to the comprehension of temporarily ambiguous sentences. *Journal of Memory and Language*, 37(1):58–93.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. Frage: Frequency-agnostic word representation. In *Advances in Neural Information Processing Systems*, volume 31.

Robert D Hawkins, Takateru Yamakoshi, Thomas L Griffiths, and Adele E Goldberg. 2020. Investigating representations of verb bias in neural language models. *arXiv preprint arXiv:2010.02375*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. `https://spacy.io`.

Harold Jeffreys. 1998. *The Theory of Probability*. Oxford University Press.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago press.

Beth Levin and Malka Rappaport Hovav. 1994. *Unaccusativity: At the syntax-lexical semantics interface*, volume 26. MIT press.

Tal Linzen and T Florian Jaeger. 2016. Uncertainty and expectation in sentence processing: Evidence from subcategorization distributions. *Cognitive science*, 40(6):1382–1411.

Maryellen C MacDonald, Neal J Pearlmutter, and Mark S Seidenberg. 1994. The lexical nature of syntactic ambiguity resolution. *Psychological review*, 101(4):676.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Aya Meltzer-Asscher, Jennifer E Mack, Elena Barbieri, and Cynthia K Thompson. 2015. How the brain processes different dimensions of argument structure complexity: Evidence from fmri. *Brain and language*, 142:65–75.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, William Brockman, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE.

James Myers. 2017. Acceptability judgments. In *Oxford Research Encyclopedia of Linguistics*. Oxford University Press.

OpenAI. 2024a. Gpt-4o system card. https://arxiv.org/abs/2410.21276.

OpenAI. 2024b. Openai o1 system card. https://arxiv.org/abs/2412.16720.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Slav Petrov and Dan Klein. 2007. Improved inference for unlexicalized parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411.

Martin J Pickering and Holly P Branigan. 1998. The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and language*, 39(4):633–651.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Laura Rimell, Stephen Clark, and Mark Steedman. 2009. Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 813–821. Association for Computational Linguistics.

Josef Ruppenhofer, Michael Ellsworth, Miriam RL Petruck, Christopher R Johnson, and Jan Scheffczyk. 2016. *FrameNet II: Extended theory and practice*. Institut für Deutsche Sprache, Bibliothek.

Rachel A Ryskin, Zhenghan Qi, Melissa C Duff, and Sarah Brown-Schmidt. 2017. Verb biases are shaped through lifelong learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 43(5):781.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, University of Pennsylvania.

Mark S Seidenberg and James L McClelland. 1989. A distributed, developmental model of word recognition and naming. *Psychological review*, 96(4):523.

Einat Shetreet, Dafna Palti, Naama Friedmann, and Uri Hadar. 2007. Cortical representation of verb processing in sentence comprehension: Number of complements, subcategorization, and thematic frames. *Cerebral Cortex*, 17(8):1958–1969.

Paul Smolensky. 1988. On the proper treatment of connectionism. *Behavioral and brain sciences*, 11(1):1–23.

Adrian Staub, Charles Clifton Jr, and Lyn Frazier. 2006. Heavy np shift is the parser's last resort: Evidence from eye movements. *Journal of memory and language*, 54(3):389–406.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. *The Penn treebank: An overview*, chapter 1. Springer.

DL Theijssen, H van Halteren, KM Fikkers, Frederike Groothoff, L van Hoof, E Sande, Jorieke Tiems, Véronique Verhagen, and PHE van der Zande. 2009. A regression model for the english benefactive alternation. In *Computational Linguistics in the Netherlands 2009*, pages 115–130. Utrecht: LOT.

John C Trueswell and Albert E Kim. 1998. How to prune a garden path by nipping it in the bud: Fast priming of verb argument structure. *Journal of memory and language*, 39(1):102–123.

John C Trueswell, Michael K Tanenhaus, and Christopher Kello. 1993. Verb-specific constraints in sentence processing: separating effects of lexical preference from garden-paths. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 19(3):528.

Cyma Van Petten and Marta Kutas. 1990. Interactions between sentence context and word frequencyineventrelated brainpotentials. *Memory & cognition*, 18:380–393.

Loan Cam Vuong and Randy C Martin. 2015. The role of lifg-based executive control in sentence comprehension. *Cognitive Neuropsychology*, 32(5):243–265.

Aaron Steven White and Kyle Rawlins. 2020. Frequency, acceptability, and selection: A case study of clause-embedding. *Glossa: a journal of general linguistics*, 5(1).

Haitong Yang, Tao Zhuang, and Chengqing Zong. 2015. Domain adaptation for syntactic and semantic dependency parsing using deep belief networks. *Transactions of the Association for Computational Linguistics*, 3:271–282.

A Context generation

For context generation, we used Open AI's ChatGPT interface to the GPT-o1 model (OpenAI, 2024b) on January 25th, 2025 with the following prompt:

I'm working on a norming task where I ask participants to produce words in sentences. I want to provide participants with a short context – just a simple phrase like "on the tennis court," "debating vaccine mandates," or "about to give birth." I will instruct them to say the first sentence that comes to mind using given a word and one of these contexts. Please help me come up with contexts by generating a CSV with two columns: "number" and "context". The "number" column should number the contexts from 1 to 1000, and the "context" column should contain 1000 different contexts, all roughly 3-4 words long. After generating this CSV, please go back and ensure that there are no duplicates.

B Sentence generation

B.1 System message

```
You are a random sentence generator,
    tasked with generating natural-
    sounding sentences like those you
    might find in a conversation, movie,
     or newspaper.
The random sentence generator should
    provide sentences with varied
    meanings, tenses, and syntactic
    structures, simulating random draws
    from sentences anywhere on the
    internet -- including in movies,
    newspapers, conversations, forums,
    etc.
I will provide you with a short context
    and a numbered list of exactly " +
    str(n_verbs_expected) + " verbs.
You must return exactly " + str(n_verbs_
    expected) + " lines, one line per
    verb in the same order, each of
    which uses the verb of the
    corresponding number in a sentence
    that might be uttered in the given
    context.
Responses do not have to include words
    from the context, but please ensure
    they are thematically related.
Please return responses in tab-delimited
     format, with the verb in the first
    column (labeled 'verb') and the
    corresponding sentence in the second
     (labeled 'sentence').
Number the sentences according to the
    numbers for each verb using the
    format '1. [sentence here]', '2. [
    some other sentence]', etc.
For instance, if I gave you the context
    'at the beach' and the verb list '1.
     give\n2. stop\n3. trip\n4. hold',
you might respond: 'give\t1. I gave the
    children a shovel and pail to make a
     sand castle.\nstop\t2. Stop!\ntrip\
    t3. While running in the surf, she
    tripped over driftwood.\nhold\t4.
    Hold onto the fishing rod for me
    while I run to the hotel!'" +
f"\n\nContext: {context_str}\n\n
```

C LLM Parser

C.1 System message

```
You are an expert linguist.
```

```
You will be given a structured JSON
    object containing multiple verb-
    sentence pairs. Each entry includes:
- id: A unique numeric identifier.
- verb: The target verb.
- sentence: The full sentence containing
     the verb.

Your task is to, for each entry:
1. Identify the clause containing the
    given verb.
2. Remove everything except the clause
    that contains the verb.
3. Within this clause, remove:
   - The subject of the clause.
   - The verb itself.
   - All optional modifiers and adjuncts
        (e.g., time, manner, or location
        expressions that are not
       required or licensed by the verb).

4. Return only the verb's selected/
    subcategorized arguments, meaning
    the arguments that the verb requires
     or licenses.

Formatting Rules:
- Each argument must be enclosed in
    square brackets: [].
- Once you have done this, label each
    bracketed argument by what kind of
    phrase/clause it is, using the
    following labels (Penn Treebank
    clause- and phrase-level tags)
    exactly: S, SBAR, ADJP, ADVP, NP, PP
    , PRT, QP, VP, WHADJP, WHAVP, WHNP,
    WHPP.
- If the verb is intransitive (has no
    arguments), return: '[intransitive]'.

- If the verb is not present or not used
     as a verb, return: 'NA'.

**Expected JSON Output Format:**
For each input entry, return a JSON
    object with:
- id (must match the input ID exactly)
- verb (must match the input verb
    exactly)
- arguments (a list of extracted
    arguments in brackets and tagged).

Example verb-sentence pairs (input) and
    the target formatted arguments (
    ouptut):
- Verb: gave, Sentence: 'She gave me a
    cake on my birthday.' -> '[me]_NP [a
     cake]_NP'
- Verb: sent, Sentence: 'We sent the
    package to the wrong address.' -> '[
    the package]_NP [to the wrong
    address]_PP'
- Verb: left, Sentence: 'He left home
    yesterday.' -> '[home]_NP'
- Verb: believe, Sentence: 'I want to
    believe that he'll be okay, but it's
     not a given.' -> '[that he'll be
    okay]_SBAR'
- Verb: paint, Sentence: 'As soon as
    they moved in they painted the house
     yellow, horrifying the neighbors.'
    -> '[the house]_NP [yellow]_ADJP'
- Verb: told, Sentence: 'They told me to
     piss off.' -> '[me]_NP [to piss off
    ]_VP'
- Verb: know, Sentence: 'I know the
    answer for sure.' -> '[the answer]_
    NP'
- Verb: complain, Sentence: 'She
    complained about the noise.' -> '[
    about the noise]_PP'

Particles should be included separately
    if they belong to the verb (e.g., '
    gave up' -> '[up]_PRT'), whereas
    prepositions should be included only
     if they are the head of a phrase
    required by the verb (e.g., '
    apologize to someone' -> '[to
    someone]_PP'). \
- Verb: threw, Sentence: 'She was so
    nervous she threw up as soon as she
    arrived.' -> '[up]_PRT'
- Verb: own, Sentence: 'And worst of all
    , he never owned up to having lied
    to us all.' -> '[up]_PRT [to having
    lied to us all]_PP'
Treat coordinated structures as a single
     structure, as in:
- Verb: give, Sentence: 'She gave me and
     my sister candy.' -> '[me and my
    sister]_NP [candy]_NP'
If the verb has no arguments (is
    intransitive), return '[intransitive
```

```
    ]', as in:
- Verb: slept, Sentence: 'I slept
    soundly.' -> '[intransitive]'

If the verb is not present in the
    sentence, or if it is not used as a
    verb, return 'NA', as in:
- Verb: chuckle, Sentence: 'Susan gave a
     quick chuckle before she turned and
     walked away.' -> 'NA' (since
    chuckle is used as a noun rather
    than a verb).

Do NOT infer missing arguments. Only
    extract what is explicitly present
    in the sentence.
Return only the extracted arguments. Do
    NOT include explanations or
    additional commentary.
Return the extracted arguments in the
    same order as they appear in input.

Example JSON input and output:
```json
{
  "input": [
    { "id": 1, "verb": "gave", "sentence
        ": "She gave me a cake on my
        birthday." },
    { "id": 2, "verb": "sent", "sentence
        ": "We sent the package to the
        wrong address." },
    { "id": 3, "verb": "left", "sentence
        ": "He left home yesterday." }
  ]
}

Example JSON output, corresponding to
    the above example JSON input: \
```json
{
  "output": [
    { "id": 1, "verb": "gave", "
        arguments": ["[me]_NP", "[a cake
        ]_NP"] },
    { "id": 2, "verb": "sent", "
        arguments": ["[the package]_NP",
         "[to the wrong address]_PP"] },
    { "id": 3, "verb": "left", "
        arguments": ["[home]_NP"] }
  ]
}

Each entry in the output **must include
    the original ID**. \
Do not change the order or skip any
    entries. \
Before returning the output, double-
    check that: \
1. Every input verb has a corresponding
    output.\n \
2. The number of entries in the output
    matches the number of inputs.\n \
3. Each response contains only extracted
     arguments in brackets with phrase/
    clause labels in the format [
    argument]_label.

Return only a well-formed JSON object.
    Do not include any additional text,
    explanations, or formatting outside
    the JSON structure.
```

D Frequent Subcategorization Frames and Examples

The 10 most common frames (summing across verb frame categories assigned by GPT-4o, the Berkeley Neural Parser, and the Stanford Parser) are listed below, with examples from our dataset. Figure 1 shows their counts per parser in our dataset (out of a total of 61,427 sentences, after exclusions).

1. NP (i.e., transitive), like *threaten* in "The ongoing crisis threatens the stability of the region."

2. NP-PP, like *reserve* in "We need to reserve resources for those who need them most."

3. PP, like *insist* in "We must insist on transparency in the recovery process."

4. SBAR (i.e., sentential complement), like *note* in "Please note that the situation is evolving rapidly."

5. Intransitive, like *arrive* in "Help finally began to arrive after days of waiting."

6. NP-SBAR, like *ask* in "I will ask my neighbors if they need help after the storm."

7. S, like *pretend* in "Don't pretend you know how to tie that knot if you really don't."

8. PP-PP, like *boast* in "I will boast about this trip to everyone I know."

9. VP, like *regret* in "We may regret not acting sooner if we don't step up now."

10. NP-ADVP, like *throw* in "I want to throw my worries aside and just ask for help."

E Comparing the GPT-4o and Berkeley Parsers to Gahl et al.

We compared the results from the GPT-4o and Berkeley parsers to Gahl et al.'s manually annotated data for the seven modal frames in Gahl et al.'s dataset: intransitives (including PPs, as in Gahl et al.; Fig. 7), transitives (i.e., NPs; Fig. 2) sentential complements (SCs; Fig. 8), particles (e.g., *clean up*; Fig. 9), particles and NPs (e.g., *pick up your clothes*; Fig. 10), nouns and sentences (e.g., *tell your mother]$_{NP}$ [(that) she's always welcome]$_{SC}$*; Fig. 11), and non-finite clauses (e.g., *he refused to try*; Fig. 12). Results of the linear models are summarized in Table 2.

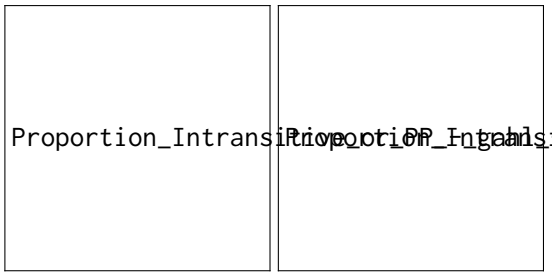

Figure 7: Evaluating the GPT-4o and Berkeley Parser's estimates of intransitivity rates by comparing them to Gahl et al.'s (2004) gold-standard dataset. To facilitate comparison, we followed Gahl et al. and counted prepositional phrase (PP) complements as intransitive for this analysis. Both automatic parsers' results were significantly correlated with the Gahl et al. frequencies, but GPT-4o (left) produced a better fit ($r^2 = .611$) than the Berkeley Parser (right; $r^2 = .339$).

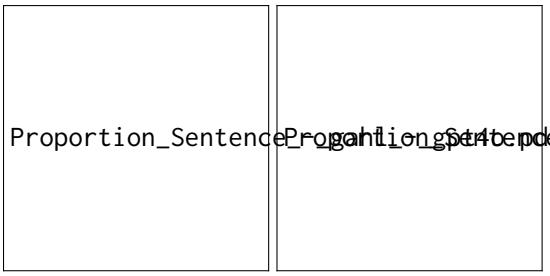

Figure 8: Comparing our estimates of sentential complement (SC; e.g., *believe [that you can do it]$_{sc}$*) frequencies per verb to Gahl et al. (2004). GPT-4o (left) accounted for slightly more variance than the Berkeley Parser (right).

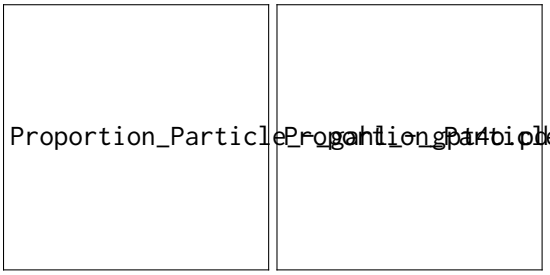

Figure 9: Comparing our estimates of particle (e.g., *clean up*) frequencies per verb to Gahl et al. (2004). GPT-4o (left) accounted for more variance than the Berkeley Neural Parser (right).

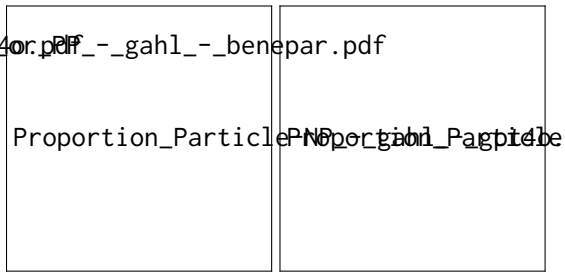

Figure 10: Comparing our estimates of particle-NP (e.g., *pick [up]$_{part.}$ [your clothes]$_{NP}$*) frequencies per verb to Gahl et al. (2004). GPT-4o (left) accounted for more variance than the Berkeley Neural Parser (right).

Figure 11: Comparing our estimates of NP-Sentence (e.g., *tell [your mother]$_{NP}$ [(that) she's always welcome]$_{SC}$*) frequencies per verb to Gahl et al. (2004). GPT-4o (left) accounted for more variance than the Berkeley Neural Parser (right).

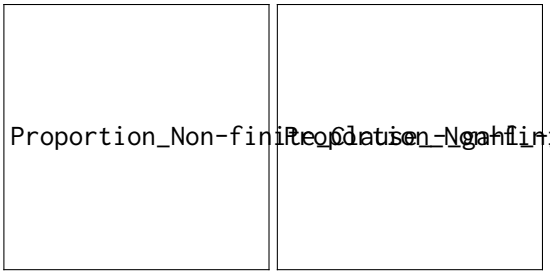

Figure 12: Comparing our estimates of nonfinite clausal complements (e.g., *he refused to try*) frequencies per verb to Gahl et al. (2004). GPT-4o (left) accounted for more variance than the Berkeley Neural Parser (right).

	GPT-4o			Berkeley		
Structure	r^2	p-value		r^2	p-value	
Transitive	.417	<.001	***	.187	<.001	***
Intransitive	.517	<.001	***	.151	<.001	***
Intrans./PP	.611	<.001	***	.339	<.001	***
Sentence	.520	<.001	***	.509	<.001	***
Particle	.109	<.001	***	.040	.002	**
NP-Sentence	.124	<.001	***	.003	.365	*n.s.*
Non-finite	.544	<.001	***	.513	<.001	***
Particle-NP	.267	<.001	***	.192	<.001	***

Table 2: Model results. For the seven modal structures in Gahl et al. (2004), we modeled their gold-standard VFF estimates as a function of our estimates from the GPT-4o and Berkeley parses. Significance codes: *n.s.*: not significant; * < .05; ** < .01; *** < .001.

F Comparisons between GPT-4o and the Stanford Parser

In the main text and preceding supplementary sections, we compared GPT-4o's performance to that of the Berkeley Neural Parser. Here, we repeat those comparisons using the Stanford Parser, which, while still widely used, reflects an earlier generation of constituency parsing models. Table 3 summarizes the results of a series of Bayesian model comparisons aiming to determine whether there was more evidence for the GPT-4o estimates or the Stanford estimates. Where there was evidence, it decisively favored GPT-4o.

Structure	**Winning Parser**	log_{10}**(BF)**	**Degree of Evidence**
Transitive	GPT-4o	15.064	Decisive
Intransitive	GPT-4o	36.518	Decisive
Sentence	GPT-4o	8.889	Decisive
Particle	Stanford	-0.350	None
NP-Sentence	GPT-4o	7.213	Decisive
Non-finite	GPT-4o	5.404	Decisive
Particle-NP	GPT-4o	9.445	Decisive

Table 3: Model comparisons. For seven structures, we compared models predicting Gahl et al.'s (2004) gold-standard VFFs using the GPT-4o or Stanford VFFs. The resulting Bayes Factors (BFs) indicate how much more evidence there was for the GPT-4o VFFs than the Stanford VFFs (positive values in log-space) or vice versa (negative). BFs were interpreted according to the Jeffreys (1998) scale. For six of the seven structures, we found "decisive" evidence that the GPT-4o estimates fit Gahl et al.'s VFFs better than the Stanford estimates. For the particle frame, there was numerically more evidence for the Stanford results, but not enough to be credible ("barely worth mentioning" on the Jeffreys scale).

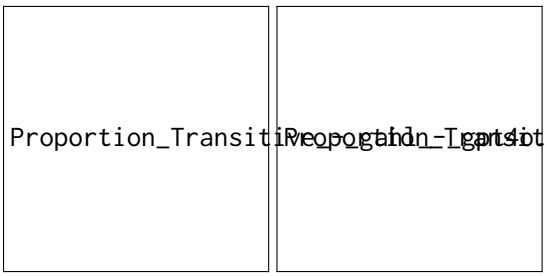

Figure 13: Evaluating the GPT-4o and Stanford Parsers' estimates of transitivity (i.e., NP frame) rates by comparing them to Gahl et al.'s (2004) gold-standard dataset. Results showed significant correlations for both parsers, but GPT-4o (left) produced a better fit ($r^2 = .417$) than the Stanford Parser (right; $r^2 = .220$).

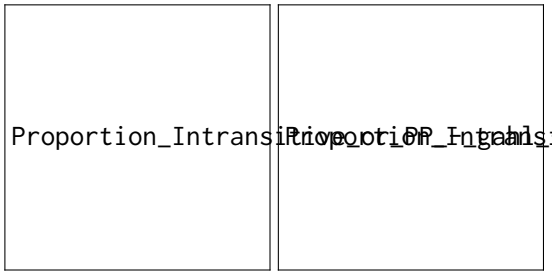

Figure 14: Comparing the GPT-4o and Stanford parsers' intransitivity rates to Gahl et al. (2004). To facilitate comparison, we followed Gahl et al. and counted prepositional phrase (PP) complements as intransitive for this analysis. GPT-4o (left) produced a better fit ($r^2 = .611$) than the Stanford Parser (right; $r^2 = .351$).

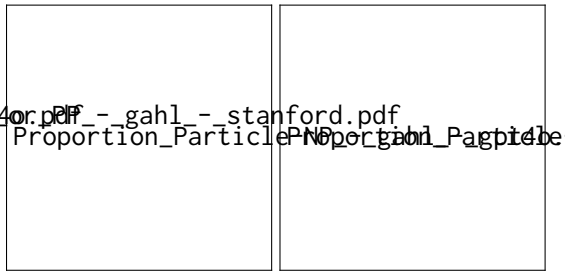

Figure 17: Comparing our estimates of particle-NP (e.g., *pick [up]$_{part.}$ [your clothes]$_{NP}$*) frequencies per verb to Gahl et al. (2004). GPT-4o (left) accounted for more variance than the Stanford Parser (right).

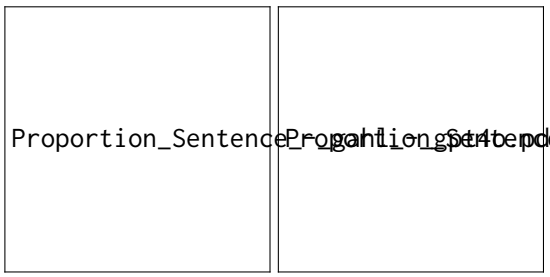

Figure 15: Comparing our estimates of sentential complement (SC; e.g., *believe [that you can do it]$_{sc}$*) frequencies per verb to Gahl et al. (2004). GPT-4o (left) accounted for more variance than the Stanford Parser (right).

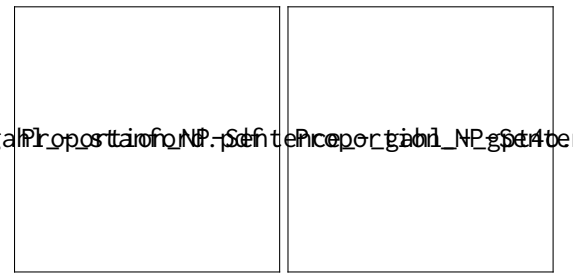

Figure 18: Comparing our estimates of NP-Sentence (e.g., *tell [your mother]$_{NP}$ [(that) she's always welcome]$_{SC}$*) frequencies per verb to Gahl et al. (2004). GPT-4o (left) accounted for more variance than the Stanford Parser (right).

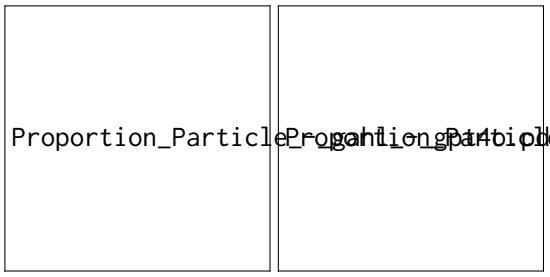

Figure 16: Comparing our estimates of particle (e.g., *clean up*) frequencies per verb to Gahl et al. (2004). GPT-4o (left) accounted for the same amount of variance as the Stanford Parser (right), although whatever success the Stanford model had here was largely driven by one verb: *sit*.

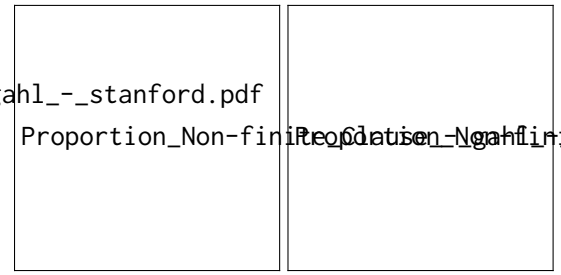

Figure 19: Comparing our estimates of nonfinite clausal complements (e.g., *he refused to try*) frequencies per verb to Gahl et al. (2004). GPT-4o (left) accounted for more variance than the Stanford Parser (right).

Structure	GPT-4o r^2	GPT-4o p-value		Stanford r^2	Stanford p-value	
Transitive	.417	<.001	***	.220	<.001	***
Intrans./PP	.611	<.001	***	.351	<.001	***
Sentence	.520	<.001	***	.433	<.001	***
Particle	.109	<.001	***	.108	<.001	***
NP-Sentence	.124	<.001	***	.004	.323	*n.s.*
Non-finite	.544	<.001	***	.001	<.001	***
Particle-NP	.267	<.001	***	.131	<.001	***

Table 4: Model results. For the seven modal structures in Gahl et al. (2004), we modeled their gold-standard VFF estimates as a function of our estimates from the GPT-4o and Stanford parsers. Significance codes: *n.s.*: not significant; * < .05; ** < .01; *** < .001.

F.1 The NP/SC Ambiguity: Stanford Comparisons

Figure 20 shows the results of comparing the relative rate of NP and SC completions from the GPT-4o and Stanford Parsers to estimates from the literature (a direct parallel to Fig. 4). The GPT-4o estimates significantly predicted the estimates from Trueswell et al. (1993) ($r^2 = .468$, $p < .001$) and Garnsey et al. (1997) ($r^2 = .390$, $p < .001$), while the Stanford Parser significantly predicted the Garnsey et al. biases ($r^2 = .386$, $p < .001$) but not the Trueswell et al. ($r^2 = .069$, $p = .075$). Bayesian model comparison of GPT-4o vs. Stanford results revealed "decisive" evidence in favor of the GPT-4o estimates when modeling Trueswell et al.'s data (log_{10}(BF)= 4.128; Jeffreys 1998), but no noteworthy evidence in favor of either parser for modeling Garnsey et al.'s data (log_{10}(BF)= .040).

Figure 20: Evaluating GPT-4o (left column) and Stanford Parser (right) estimates of NP bias for NP/SC verbs. We compared our estimates to those from two prior studies: Trueswell et al. (1993; top row) and Garnsey et al. (1997; bottom). Both parsers significantly predicted the Garnsey et al. results, but only GPT-4o significantly predicted Trueswell et al.'s data.

F.2 Intransitivity Estimates: Stanford Comparisons

Here we report the analyses in Section 3.5, but comparing GPT-4o to the Stanford Parser. Consistent with our findings above, Fig. 21 shows that excluding PPs from intransitivity counts again resulted in worse fits for both parsers, reducing r^2 from .611 to .517 for the GPT-4o parser (left column) and from .351 to .023 for the Stanford Parser (right).

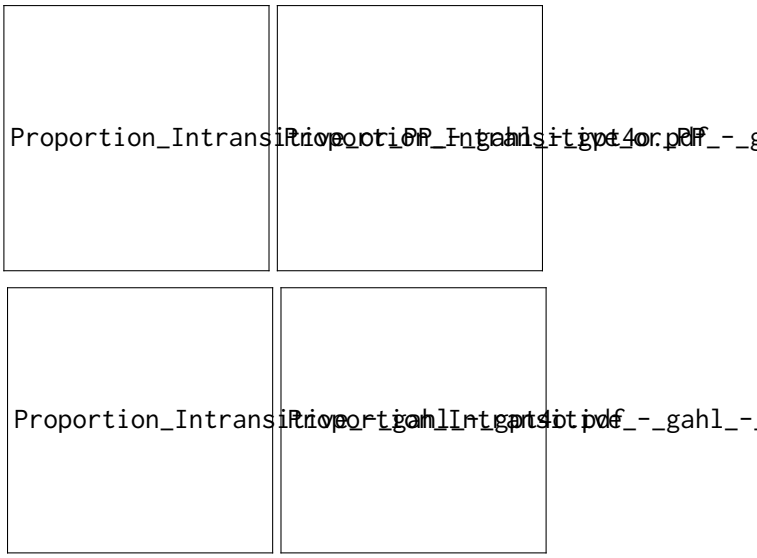

Figure 21: Evaluating the effect of including PP frames in intransitivity counts. For both the GPT-4o (left) and Stanford (right) parsers, including PPs (as Gahl et al. (2004) did) produced a better fit (top) to Gahl et al.'s estimates than excluding them (bottom), meaning that Gahl et al. lost meaningful information by collapsing these categories.

F.3 The Dative Alternation: Stanford Comparisons

To evaluate the accuracy of these estimates, we compared them to the relative preference ratings reported in Hawkins et al. (2020) for the 150 verbs our studies shared in common. Many verbs had no DO completions at all (the row of dots at the bottom of both panels in Fig. 6), consistent with Hawkins et al.'s inclusion of non-alternating PO verbs. As before, we performed a Bayesian model comparison, which revealed "decisive" evidence (log_{10}(BF)= 3.585; Jeffreys 1998) in favor of the GPT-4o results over the Stanford results.

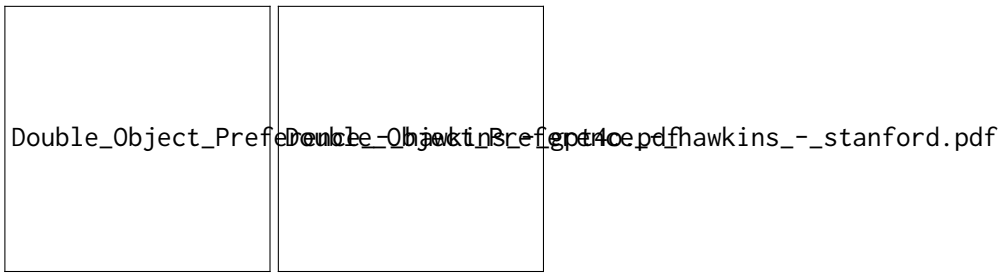

Figure 22: Evaluating the GPT-4o (left) and Stanford (right) Parsers' estimates of DO/PD bias for dative verbs. We compared estimates to Hawkins et al.'s (2020) relative preference ratings. Many of the verbs they included are non-alternating (e.g., *whisper*), and for the majority of these neither GPT-4o nor the Stanford parser identified any DO completions. To avoid infinities, we set proportions of 0 (no instances) or 1 to .001 and .999, respectively, capping log-odds values at ± 6.907 (the row of black dots at the bottom of each plot). Both parsers significantly predicted preference ratings, accounting for comparable amounts of variance (GPT-4o $r^2 = .195$; Stanford $r^2 = .229$), although Bayesian model comparison revealed "decisive" evidence in favor of the GPT-4o estimates.