Characterizing English Preposing in PP constructions

Christopher Potts
Stanford University

Abstract The English Preposing in PP construction (PiPP; e.g., *Happy though/as we were*) is extremely rare but displays an intricate set of stable syntactic properties. How do people become proficient with this construction despite such limited evidence? It is tempting to posit innate learning mechanisms, but present-day large language models seem to learn to represent PiPPs as well, even though such models employ only very general learning mechanisms and experience very few instances of the construction during training. This suggests an alternative hypothesis on which knowledge of more frequent constructions helps shape knowledge of PiPPs. I seek to make this idea precise using model-theoretic syntax (MTS). In MTS, a grammar is essentially a set of constraints on forms. In this context, PiPPs can be seen as arising from a mix of construction-specific and general-purpose constraints, all of which seem inferable from experience.

Keywords: unbounded dependency constructions, large language models, corpus linguistics, model-theoretic syntax, stimulus poverty arguments

1 Introduction

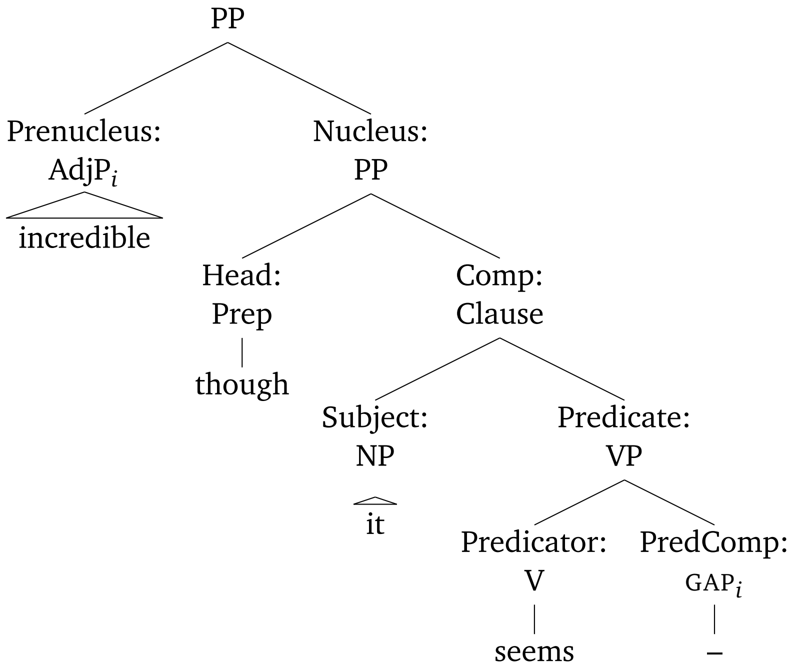
The examples in (1) illustrate what Huddleston and Pullum (2002; *CGEL*) call the English Preposing in PP construction (PiPP):

(1) a. Happy though we were with the idea, we decided not to pursue it.
  b. Brilliant linguists though they were, they just couldn't figure it out.
  c. Brilliant as they seemed, they just couldn't figure it out.

This is a draft of a planned submission to a Festschrift volume, *English language and linguistic theory: A tribute to Geoff Pullum*. My thanks to Richard Futrell, Julie Kallini, Kyle Mahowald, Isabel Papadimitriou, and Brett Reynolds for extremely helpful discussion. And a special thanks to Geoff for all his guidance and support over the years. Geoff's research reflects the best aspects of linguistics, and of scientific inquiry in general: it is open-minded, rigorous, empirically rich, methodologically diverse, and carefully and elegantly reported. In all my research and writing, Geoff is an imagined audience for me, and this has helped push me (and, indirectly, my own students) to try to live up to the incredibly high standard he has set. The code and data for this paper are available at https://github.com/cgpotts/pipps.

On the *CGEL* analysis (in chapter 7, 'Prepositions and prepositional phrases', by Geoffrey K. Pullum and Rodney Huddleston), PiPPs are PPs headed by the preposition *though* or *as*, and the preposed predicational phrase enters into an unbounded dependency relationship with a gap inside a complement clause. The following is *CGEL*'s core constituency analysis (p. 633):

(2)



The *CGEL* description of PiPPs focuses on three central characteristics of the construction: (1) it is limited to *though* and *as*, with *as* optionally taking on a concessive sense only in PiPPs; (2) it can target a wide range of phrases; and (3) it is a long-distance dependency construction, as seen in (3).

(3) a. Happy though/as we know that they would think that others would be with the idea, . . .

b. Brilliant linguist though/as his friends would testify that his colleagues say that he is, . . .

In my first year in graduate school, Geoff Pullum taught a mathematical linguistics course (Spring 2000 quarter) that drew on his ongoing work with Rodney Huddleston on *CGEL*. At one of the meetings, Geoff challenged the class to find attested cases of PiPP constructions spanning finite-clause boundaries, and he offered a $1 reward for each example presented to him by the next meeting.

At the time, the best I could muster was (4). These are single-clause PiPPs, but Geoff awarded $0.05 in cash in recognition of my habit of collecting interesting examples.

(4) a. "Hungry though I am for life as the next fellow sometimes I think that lying under the ground there would not be such a bad thing." (Joseph Epstein. *With My Trousers Rolled*, p. 281.)

b. "Laudable though Potter's ends were, and wonderfully perverse his means, ..." (Joseph Epstein. *A Line out for a Walk*, p. 47.)

I kept an eye out for long-distance PiPPs, but this mostly turned up infinitival cases like (5).

(5) a. "Roland felt a huge irritability mounting inside himself, mild though he knew himself to be, . . . " (A.S. Byatt, *Possession*, p. 105.)

b. ". . . workmen with whom one exchanges salutations when one passes them in the streets of the capital, engaged as they tend to be in reexcavating the same stretch of street that they were digging up only a few weeks before." (John Lanchester, *The Debt to Pleasure*, p. 151)

It was not until 2002 that I found (6a). My triumphant message to Geoff is given in Appendix A. This was sadly too late to help with *CGEL*, and Geoff awarded no cash prize. However, I am proud to report that the example is cited in Pullum 2017. It appears alongside (6b), which was found by Mark Davies in 2009. Mark apparently heard about Geoff's quixotic PiPP hunt and tracked down at least one case in CoCA (Davies 2008). In 2011, Geoff finally found his own case, (6c), which is noteworthy for being from unscripted speech.

(6) a. "Although he sometimes retreated to a stance of pure practicality, Feynman gave answers to these questions, philosophical and unscientific though he knew they were." (James Gleick, *Genius: The Life and Science of Richard Feynman*, p. 13.)

b. "Good though he knew it was, . . . " (CoCA)

c. "Unpopular though I can well see that it might be, . . . " (Radio 4, April 12, 2011. Story on the European Court of Human Rights.)

I believe Geoff's motivations for issuing the PiPP challenge were twofold. First, PiPPs embody a central insight: linguistic phenomena can be both incredibly rare and sharply defined. Second, he was hoping we might nonetheless turn up attested examples to inform the characterization of PiPPs that he and Rodney were developing for *CGEL*. My sense is that, while Geoff is very happy to make use of invented examples, he feels that a claim isn't secure until it is supported by independently attested cases.[1] This aligns with how he reported example (6c) to me: "At last, confirmation of the unboundedness from speech!" (Geoff's email message is reproduced in full in Appendix B.)

Ever since that turn-of-the-millenium seminar, PiPPs have occupied a special place in my thinking about language and cognition. Because of Geoff's challenge, PiPPs are, for me, the quintessential example of a linguistic phenomenon that is both incredibly rare and sharply defined. With the present paper, I offer a deep dive on the construction using a mix of linguistic intuitions, large-scale corpus resources, large language models, and model-theoretic syntax. My goal is to more fully understand what PiPPs are like and what they can teach us.

My investigation centers around corpus resources that are larger than the largest Web indices were in 1999–2000 (Section 2).[2] These corpora provide a wealth of informative examples that support and enrich the *CGEL* description of PiPPs (Section 3). They also allow me to estimate the frequency of PiPPs (Section 4). The overall finding here is that PiPPs are indeed incredibly rare: I estimate that under 0.03% of sentences in literary text contain the construction (and rates are even lower for general Web text). By comparison, about 12% of sentences include a restrictive relative clause. Nonetheless, and reassuringly, this corpus work does turn up naturalistic PiPP examples in which the unbounded dependency crosses a finite-clause boundary; were Geoff's offer still open, I would stand to earn $58 (see Appendix E).

The vanishingly low frequency of PiPP's raises the question of how people manage to acquire and use the construction so systematically. It's very hard to imagine that these are skills honed entirely via repeated uses or encounters with the construction itself. In this context, it is common for linguists to posit innate learning mechanisms – this would be the start of what Pullum and Scholz (2002) call a *stimulus poverty argument*, based in this case on the notion that the evidence underdetermines the final state in ways that can only be explained by innate mechanisms. Such mechanisms may well be at work here, but we should ask whether this is truly the only viable account.

To probe this question, I explore whether present-day large language models (LLMs) have learned anything about PiPPs. Building on methods developed by Wilcox et al. (2023), I present evidence that GPT-3 models (Brown et al.

[1] Pullum (2017) criticizes the extremes of "corpus fetishism" and "intuitional solipsism"and argues for a wide-ranging approach to evidence in linguistics (see also Pullum 2007b). For a lively summary of this view, see Pullum 2009: §5.

[2] The C4 corpus I use in this paper has 365M documents in the en section. According to Sullivan (2005), the largest Web indices in 1999 had 200M pages, though Google announced in June 2000 that it had reach 500M.

2020) have excellent command of the core properties of PiPPs identified in *CGEL* and summarized in Section 3. These models are exposed to massive amounts of text as part of training, but they are in essentially the same predicament as humans are when it comes to learning about PiPPs: PiPPs are exceedingly rare in their training data. Importantly, these models employ only very general purpose learning mechanisms, so their success indicates that specialized innate learning mechanisms are not strictly necessary for becoming proficient with PiPPs (for discussion, see Dupoux 2018, Wilcox et al. 2023, Warstadt and Bowman 2022, Piantadosi 2023, Frank 2023a,b).

As an alternative account, I argue that, for LLMs and for humans, PiPPs arise from more basic and robustly supported facts about English. To begin to account for this capacity, I develop a model-theoretic syntax (MTS; Rogers 1997, 1998, Pullum and Scholz 2001, Pullum 2007a, 2020) account in which PiPP's follow from a mix of mostly general patterns and a few very specific patterns (Section 6). My central claim is that this MTS account is a plausible basis for explaining how PiPPs might arise in a stable way even though they are so rare.

2 Corpus resources

The qualitative and quantitative results in this paper are based primarily in examples from two very large corpus resources: BookCorpusOpen and C4.

2.1 BookCorpusOpen

This is a collection of books mostly or entirely by amateur writers. The original BookCorpus was created and released by Zhu et al. (2015), and it formed part of the training data for a number of prominent LLMs, including BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019), and GPT (Radford et al. 2018).[3]

Bandy and Vincent (2021) offer a deep investigation of BookCorpus in the form of an extensive Datasheet (Gebru et al. 2018) with commentary. This provides important insights into the limitations of the resource. For instance, though BookCorpus contains 11,038 book files, Bandy and Vincent find that it contains only 7,185 unique books. In addition, they emphasize that the corpus is heavily skewed towards science fiction and what everyone in this literature refers to euphemistically as 'Romance'.

Zhu et al. stopped distributing BookCorpus some time in late 2018, but a version of it was created and released by Shawn Presser as BookCorpusOpen.[4]

[3] The 'Books' corpora included in the training data for GPT-2 (Radford et al. 2019) and GPT-3 (Brown et al. 2020) seem to be different.

[4] https://github.com/soskek/bookcorpus

BookCorpusOpen addresses the issue of repeated books in the original corpus but seems to have a similar distribution across genres. This is the corpus of literary texts that I use in this paper. It consists of 17,688 books. The NLTK `TreebankWordTokenizer` yields 1,343,965,395 words, and the NLTK Punkt sentence tokenizer (Kiss and Strunk 2006) yields 90,739,117 sentences.

2.2 C4

C4 is the Colossal Clean Crawled Corpus developed by Raffel et al. (2020). Those authors did not release the raw data, but rather scripts that could be used to recreate the resource from a snapshot of the Common Crawl.[5] Dodge et al. (2021) subsequently created and released a version of the corpus as C4, and explored its contents in detail. Overall, they find that C4 is dominated by mostly recent texts from patent documents, major news sources, government documents, and blogs, along with a very long tail of other sources.

Dodge et al.'s discussion led me to use the `en` portion of their C4 release. This is the largest subset focused on English. The steps that were taken to create the EN.CLEAN and EN.NOBLOCKLIST subsets seemed to me to create a risk of losing relevant examples, whereas my interest is in seeing as much variation as possible. The `en` subset of C4 contains 365M documents (156B tokens). I tokenized the data into sentences using the the NLTK Punkt sentence tokenizer, which yields 7,546,154,665 sentences.

3 English Preposing in PP constructions

This section reviews the core characterization of PiPPs developed in *CGEL*. Examples from C4 are marked `C`, and those from OpenBooks with `B`. To find these examples, I relied on ad hoc regular expressions and the annotation work reported in Section 4. At a certain point, I realized I had annotated enough data to train a classifier model. This model is extremely successful (nearly perfect precision and recall on held-out examples) and so it turned out to be a powerful investigative tool. This model is described in Appendix D. I used it in conjunction with regexs to find specific example types.

3.1 Prepositional head restrictions

Perhaps the most distinctive feature of PiPPs is that they are limited to the prepositional heads *though* and *as*:

[5] https://commoncrawl.org

(7) a. ^{C}That disaster, bad as it was, would be a pinprick compared to what could happen if Line 5 broke.

b. ^{B}Young though he was, he deserved an explanation for why his life had been turned upside down.

As observed in *CGEL*, even semantically very similar words do not participate in the construction:

(8) a. That disaster, although/while it was bad, . . .

b. *That disaster, bad although/while it was, . . .

Another peculiarity of PiPPs is that *as* can take on a concessive reading that it otherwise lacks. For example, (9a) invites an additive reading of *as* that is comparable to (9b).

(9) a. Happy as we were with the proposal, we adopted it.

b. As we were happy with the proposal, we adopted it.

By contrast, the concessive context of (10) means that, whereas the PiPP is fine, the non-PiPP variant seems pragmatically contradictory because the concessive reading of *as* is unavailable.

(10) a. Happy as we were with the proposal, we couldn't adopt it.

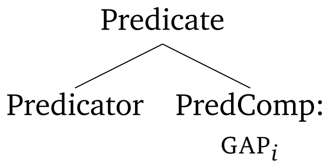
b. $^{\#}$As we were happy with the proposal, we couldn't adopt it.

It seems unlikely that we will be able to derive the prepositional-head restrictions from deeper syntactic or semantic properties. First, PiPPs don't generalize to other semantically similar concessive markers like *although* and *while*. Second, the primary distributional difference between *though* and these other candidate heads is that *though* has a wider set of parenthetical uses (*They said, though/*although, that it was fine*). However, the PiPP use is not a parenthetical one. Third, even if invoking the parenthetical uses of *though* seemed useful somehow, it would likely predict that *as* does not participate in the construction, since *as* lacks the relevant parenthetical uses. Fourth, PiPPs license an otherwise unattested concessive reading of *as*. However, fifth, PiPPs are not invariably concessive, as we see from the additive readings of *as*-headed cases. These facts seem to indicate that the prepositional-head restrictions are idiomatic and highly construction-specific.

3.2 Gap licensing

While the prepositional head restrictions in PiPPs are likely construction-specific, many properties of PiPPs do seem to follow from general principles. For example, I would venture that any Predicator that takes a PredComp in the sense of (2) can host the gap in a PiPP. Here is the relevant configuration from (2):

(11)



Here are some examples that help to convey the diversity of PiPP Predicators (which are in bold):

(12) a. ^{B}I admire the tenacity, useless though it **is**.

b. ^{C}Clay, her best surface, mitigates the flaws in her game to some extent but lovely and talented as she was and is, Ana simply never was as good as many people seem to think she was.

c. ^{C}Crushing though it **seems** innovation is a hard game requiring confidence passion and experience by the bucket load and tenacity.

d. ^{C}That's what's great about these modern techniques, cliched though spherification has **become**.

e. ^{C}Strange though it **feels** to say it and strange it may be to hear it, knowing that I'm going to die feels liberating.

f. ^{B}Precarious though they **looked**, they were actually quite solid, a formation from once-buried strata now exposed to open air.

g. ^{B}Ridiculous though it **sounds**, tis true.

h. ^{B}Their armour, strong though it **appeared**, was brittle, and no match for the strong steel of the lokchangs imperial blades.

i. ^{C}But something about that recipe nags me still, perfect though it **tastes**.

Other predicational constructions seem clearly to license PiPPs as well. Some invented examples:

(13) a. Busy as/though they kept us, I was quite bored.

b. Clean though/as they wiped the table, I still worried about germs.

Thus, I venture that any local tree structure with Predicator and PredComp children (as in (11)) is a potential target for a PiPP gap.

PiPP gaps can also be VP positions:

(14) a. $^{\mathrm{B}}$But try as he might, he couldn't quiet his racing thoughts.

b. $^{\mathrm{B}}$Struggle though he might, her grip on his hands was simply too strong.

c. $^{\mathrm{B}}$But somehow there was always a horizon and beyond it I could not see, peer though I did.

The *try as/though X might* locution is extremely common by the standards of PiPPs. There are at least 870 of them in BookCorpusOpen, 861 of which are *as*-headed. Examples like (14c) are less common but still relatively easy to find.

These non-predicational PiPP gaps can be assimilated to the others if we assume that the fronted constituent is abstractly a property-denoting expression and so has the feature PredComp. Constituents with other semantic types are clearly disallowed:

(15) a. Though Sandy saw the movie,

b. See the movie though Sandy did, . . . VP

c. *The movie though Sandy saw, . . . Direct object

d. *See/Saw though Sandy (did) the movie, . . . Verb

CGEL briefly discusses adverbial PiPP gaps as well (p. 635). Here are two such cases:

(16) a. $^{\mathrm{B}}$I'm debating going with something else with more yardage, much though I want this to be in cashmere,

b. $^{\mathrm{C}}$Hard though I looked, I didn't see any plants with unusual markings on the outer segments.

These cases seem not to satisfy the generalization that the preposed element is property denoting. It is also hard to determine what is licensing the gaps; (16b) could involve a head–complement relationship between *look* and *hard*, but there seems not to be such a relationship for *much* in (16a).

3.3 A diverse range of preposable predicates

PiPPs also permit a wide range of predicational phrases to occupy the preposed position (the Prenucleus in (2)). Here is a selection of examples:

(17) a. ^{B}**Dark, gloomy, and dangerous** though it might be, our town square was a center of admiration throughout the universe

b. ^{B}His ears were still stinging from her words as from the lashes of a whip, **kindly spoken** though they were.

c. ^{B}**Tempted to run** though he was, Will stood his ground.

d. ^{B}The intervening years, **few** though they might be, have worked their inevitable magic.

e. ^{B}This child who demanded her maternal love, **withered thing** though it was.

I would hypothesize that any phrase that can be a predicate is in principle possible as the preposed element in a PiPP. However, there are two important caveats to this, which I turn to now.

3.3.1 Adverbial *as* modification

CGEL notes that, "With concessive *as* some speakers have a preposed predicative adjective modified by the adverb *as*" (p. 634). This version of the construction is very common in the datasets I am using:

(18) a. ^{C}As spectacular as his career was, what Ali stood for as a man made the biggest impression on me.

b. ^{C}As fun as those digital adventures are, as determined as digital heroes are, they both pale in comparison with what God has done and is doing.

c. ^{B}As nervous as she was, she was still enjoying the view.

d. ^{B}As frightening as that fall was, there was something very freeing about it.

In addition, the following may be a case in which the *as*. . . *as* version of the construction has an additive rather than a concessive sense:

(19) ^{B}As sensitive as she was, she was aware of the gesture, and paused.

Here, the author seems to use the PiPP to offer rationale; a concessive reading would arise naturally if the continuation said *she was unaware of the gesture*.

In these cases, there is a mismatch between the preposed constituent and what could appear in the gap site, since this kind of *as* modification is not permitted in situ; examples like (20a) and (20b) work only on a reading meaning 'equally fast', which is quite distinct from the PiPP (20c).

(20) a. They are as fast, . . .

b. As they are as fast, . . .

c. As fast as they are, . . .

These PiPP variants superficially resemble equative comparative constructions of the form *X is as ADJ as Y*, and they are united semantically in being restricted to gradable predicates. However, the meanings of the two seem clearly to be different (*CGEL*, p. 634). In particular, whereas (20c) seems to assert that they were fast (probably in order to concede this point), examples like *Kim is as fast as Sandy is* do not entail speediness for Kim or Sandy, but rather only compare two degrees (Kennedy 2007). Thus, it seems that the *as. . . as* form is another construction-specific fact about PiPPs, though the adverbial *as* seems to have a familiar degree-modifying sense.

3.3.2 Missing determiners

When the preposed predicate is a nominal, it typically has no determiner (*CGEL*, p. 634):

(21) a. ^{B}That's why I threw in my lot with you, bloody usurping sod though you are.

b. ^{C}Macbeth, great warrior though he is, is ill equipped for the psychic consequences of crime.

c. ^{B}He had the time to discover that his mind, soldier's though it was, burned brighter than most, . . .

d. ^{B}You weren't enjoying our meetings at all, relatively short ones though they were.

e. ^{B}Sweet succor though such a death would be, . . .

In all these cases, the non-preposed version requires an indefinite determiner:

(22) a. Though you are a bloody usurping sod, . . .

b. *Though you are bloody usurping sod, . . .

(23) a. Though it was a soldier's, . . .

b. *Though it was soldier's, . . .

Conversely, retaining the determiner in the PiPP seems to be marked. However, while I have not found examples of preposed determiners in preposed NPs in

PiPP constructions, they do sound relatively okay to me, and so I have marked them with a mark • to simply convey that they are unattested:[6]

(24) a. •A bloody usurping sod though/as you are, …

b. •A solider's mind though/as it was, …

The option (or requirement) to drop the determiner in the preposed phrase seems like another construction-specific aspect of PiPPs.

3.4 Modifier stranding

In PiPPs, the entire complement to the Predicator can generally be preposed. However, it is common for parts of the phrase to be left behind, even when they are complements to the head of the PredComp phrase (*CGEL*, p. 634):

(25) a. ^{B}His wilderness-bred ears were keener even than the ears of techotl, whetted though these were by a lifetime of warfare in those silent corridors.

b. ^{B}Impatient though they were to get on, they slowed their pace …

c. ^{B}But even so, difficult though it might be for you to believe, …

d. ^{B}The decibels she employed in that one word, spoken as it was both aloud and with telepathy, pounded the hell out of his eardrums and shattered all the bottles on the bar.

In these situations, the fronted element must include the head of the predicative phrase; parts of the embedded modifier cannot be the sole target:

(26) a. *For you to believe though it might be difficult, …

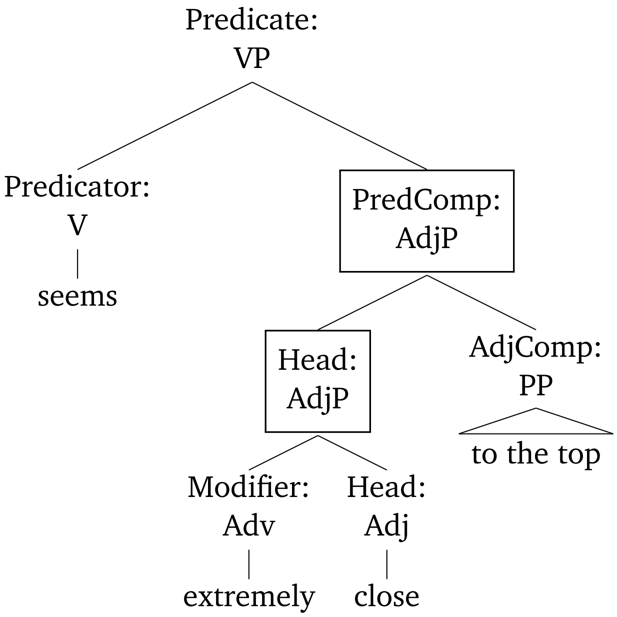
b. *By a lifetime of warfare though these were whetted, …

c. *Get on though they were impatient to, …

The generalization seems to be that the preposed element needs to a phrasal head of the PredComp phrase. For example, in (27), both the PredComp:AdjP and Head:AdjP nodes are potential targets, but the AdjComp:AdjP is not (nor is the non-phrasal Head:Adj):

[6] I feel compelled to note that I do know this is the original historical meaning of *, and that generative grammarians strengthened the meaning of * from 'unattested' to 'impossible'.

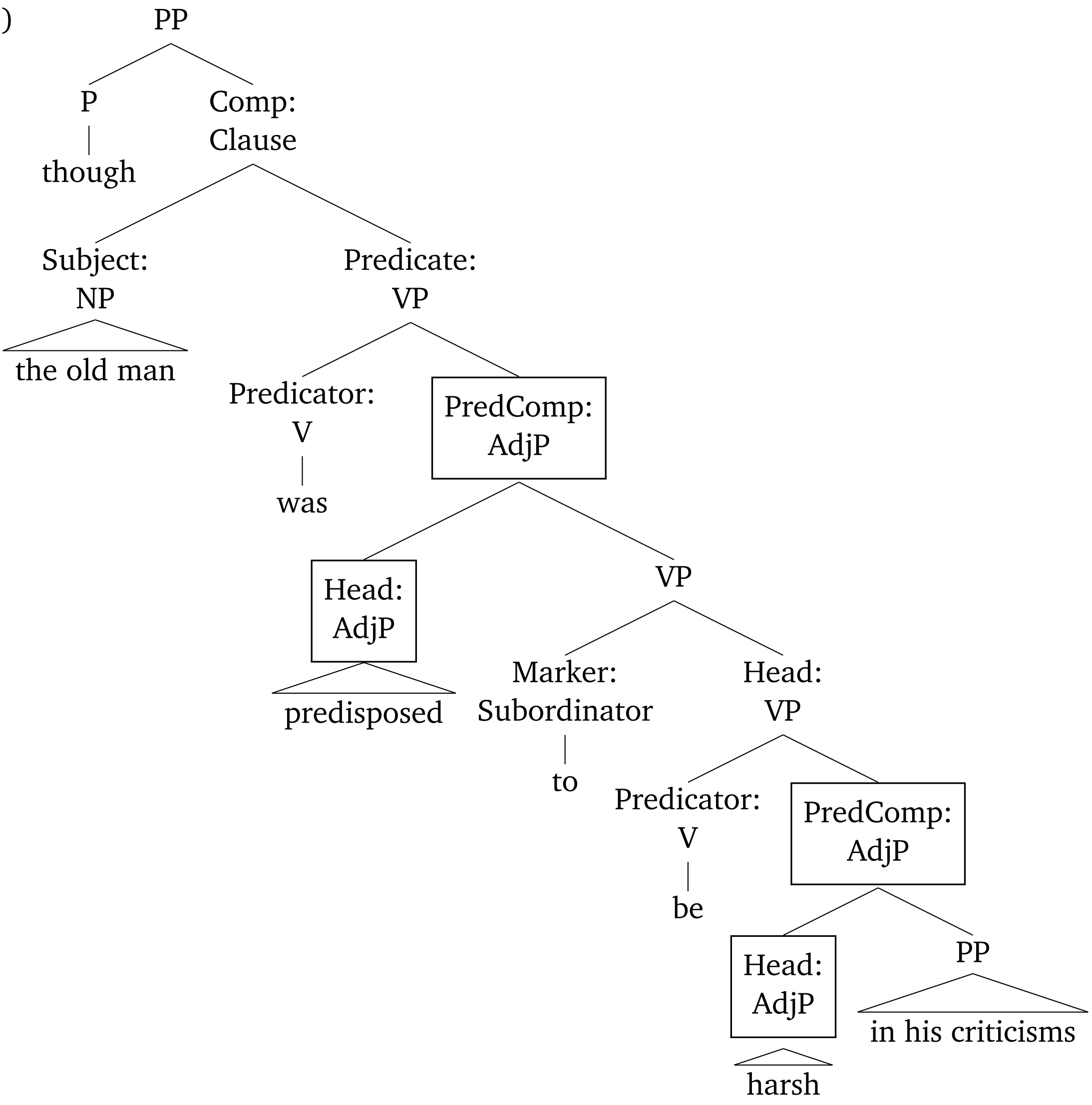
(27)



On this approach, the ungrammatical cases in (26) are explained: their gap sites do not match (27). Where there is such a match, the examples are actually fine (assuming independent constraints on unbounded dependencies are satisfied). For example, (28) contains four local trees in which a Predicator and PredComp are siblings, and in turn there are four ways to form the PiPP:[7]

[7] This example is modeled on an BookOpenCorpus case: *Harsh though the old man was predisposed to be, even his caviling nature found little to quibble about* . . . .

(28)



a. [Harsh] though the old man was predisposed to be in his criticisms,

b. [Harsh in his criticisms] though the old man was predisposed to be,

c. [Predisposed] though the old man was to be harsh in his criticisms,

d. [Predisposed to be harsh in his criticisms] though the old man was,

3.5 Long-distance dependencies

It is common for PiPPs to span infinitival clause boundaries, as in (5) and (29).

(29) a. ^{B}Guilty though I believe Mars to be,

b. ^{B}The call to power, to salvation, and false though he knew it to be, it had gained strength with every rising.

As discussed in Section 1, the initial impetus for this project was the question of whether we could find naturalistic examples of spanning finite-clause boundaries. Intuitively, these examples seem natural, but they are incredibly rare in actual data. However, they can be found. Here are two such cases:

(30) a. ^{B}Honourable though I am sure his intentions were, he betrayed you, Ruben.

b. ^{B}Eriks reassurance, heart-felt though she knew it was, did little to ease her anxiety over the impending day.

Appendix E contains all of the examples of this form that I have found. All are from written text. Geoff's example (6c) is a spoken example.

It is natural to ask whether PiPPs are sensitive to syntactic islands. This immediately raises broader questions of island sensitivity in general (Postal 1998, Hofmeister and Sag 2010). I leave detailed analysis of this question for another occasion. Suffice it to say that I would expect PiPPs to be island sensitive to roughly the same extent as any other unbounded dependency construction.

3.6 Discussion

The following seeks to summarize the characterization of PiPPs that emerges from the above *CGEL*-based discussion:

1. PiPP heads are limited to *though* and *as*, and PiPPs are the only environment in which *as* can take on a concessive reading.
2. Any complement *X* to a Predicator, or one of *X*'s phrasal heads, can in principle be a PiPP gap.
3. The Preposed element can be any property-denoting expression (and even an adverbial in some cases), and PiPPs show two idiosyncrasies here: gradable preposed elements can be modified by an initial adverbial *as*, and the expected determiner on preposed nominals is (at least usually) missing.
4. PiPPs are unbounded dependency constructions.

What sort of evidence do people (and machines) get about this constellation of properties? The next section seeks to address this question with a frequency analysis of PiPPs.

4 Corpus analyses

The goal of this section is to estimate the frequency of PiPPs in usage data.

4.1 Materials

I rely on the corpora described in Section 2, which entails a restriction to written language. In addition, while C4 is a very general Web corpus, BookCorpusOpen is a collection of literary works. Intuitively, PiPPs are literary constructions, and so using BookCorpusOpen will likely overstate the rate of PiPPs in general texts, and we can expect that rates of PiPPs are even lower in spoken language. Overall, though, even though I have chosen resources that are biased in favor of PiPPs, the central finding is that they are vanishingly rare even in these datasets.

4.2 Methods

PiPPs are infrequent enough in random texts that even large random samples from corpora often turn up zero cases, and thus using random sampling is noisy and time consuming. To get around this, I employ the following procedure for each of our two corpora C, both of which are parsed at the sentence level:

1. Extract a subset of sentences M from C using a very permissive regular expression. We assume that M contains *every* PiPP in all of C. The regex I use for this is given in Appendix C.
2. Sample a set of S sentences from M and annotate them by hand.
3. To estimate the overall frequency of sentences containing PiPPs and get a 95% confidence interval, use bootstrapped estimates based on S:
   a. Sample 100 examples B from S with replacement and use these samples to get a count estimate $\tilde{c} = (p/100) \cdot |M|$, where p is the number of PiPP-containing cases in B.
   b. Repeat this experiment 10,000 times and use the resulting $\tilde{c}$ values to calculate a mean $\hat{c}$ and 95% confidence interval.
4. By assumption 1, $\hat{c}$ is the same as the estimated number of cases in the entire corpus C. Thus, we can estimate the percentage of sentences containing a PiPP as $\hat{c}/|C|$.

4.3 Frequency estimates

BookCorpusOpen For BookCorpusOpen, we begin with $90,739,117$ sentences. The regex in Appendix C matches $5,814,960$ of these sentences. I annotated $1,000$ of these cases, which yielded 5 annotated examples. This gives us an estimate of $(5/1000) \cdot 5,814,960 = 29,075$ examples in all of BookCorpusOpen. The bootstrapping procedure in step 3 above yields an estimated count of $29,249 \pm 761$, which in turn means that roughly 0.0322% of sentences in BookCorpusOpen contain a PiPP.

C4 For C4, we begin with $7,546,154,665$ sentences. The permissive regex matches $540,516,902$ of them. I again annotated $1,000$ sentences, which identified 4 positive cases. This gives us an estimate of $(4/1000) \cdot 540,516,902 = 2,162,068$, which is very close to the bootstrapped estimate of $2,108,556 \pm 63,370$, which says that roughly 0.0279% of sentences in C4 contain a PiPP. This is lower than the BooksCorpusOpen estimate, which is consistent with the intuition that PiPPs are a highly literary construction (C4 consists predominantly of prose from non-literary genres; Section 2.2).

4.4 Discussion

The frequency estimates help to confirm that PiPPs are extremely rare constructions, present in only around 0.03% of sentences.

To contextualize this finding, I annotated 100 randomly selected cases from C4 for whether or not they contained restrictive relative clauses. I found that 12/100 cases (12%) contained at least one such relative clause. This leads to an estimate of $905,538,559$ C4 sentences containing restrictive relative clauses, compared with $2,215,038$ for PiPPs. These are very different situations when it comes to inferring the properties of these constructions.

How do these numbers compare with human experiences? It is difficult to say because estimates concerning the quantity and nature of the words people experience vary greatly. Gilkerson et al. (2017) estimate that children hear roughly 12,300 adult words per day, or roughly 4.5M words per year. Other estimates are higher. Drawing on analyses by Hart and Risley (1995), Wilcox et al. (2023: §6.2) estimate that "a typical child in a native English environment" hears roughly 11M words per year. Frank (2023a) offers a higher upper bound for people who read a lot of books: perhaps as many as 20M words per year.

At the time Geoff issued his PiPP challenge, I was 23 years old, and I was excellent at identifying and using PiPPs, if I do say so myself. The above suggests that I had experienced 100M–460M words by then. Assuming 12 words per sentence on average, and using our rough estimate of 0.03% as the percentage of PiPP-containing sentences, this means that I had heard between 2,500 and 11,500 PiPPs in my lifetime, compared with 1M–4.6M sentences containing restrictive clauses. Is 2.5K–11.5K encounters sufficient for such impressive proficiency? I am not sure, but it seems useful to break this down into a few distinct subquestions.

In Section 3, I reviewed the *CGEL* account of PiPPs. Some of the properties reviewed there seem highly construction-specific: the prepositional-head restrictions (Section 3.1), the quirky adverbial *as* appearances (Section 3.3.1), and the missing determiners (Section 3.3.2). For these properties, 2.5K–11.5K may be sufficient for learning. However, it seems conceptually like this holds only if we introduce an inductive bias: the learning agent should infer that the attested cases exhaust the range of possibilities in the relevant dimensions, so that, for example, the absence of *although*-headed PiPPs in the agent's experience leads the agent to conclude that such forms are impossible.

We need to be careful in positing this inductive bias, though. Consider the generalization that any predicate is preposable (Section 3.3). This seems intuitively true: I presented attested PiPPs with a wide range of preposed phrases. However, the attested cases cannot possibly cover what is possible; even 11.5K examples is tiny compared to the number of licit two-word adverb–adjective combinations in English, and of course preposed phrases can be longer than two words. Thus, the learning agent seemingly needs to venture that the set of attested cases is not exhaustive. Here, experience needs to invite a generalization that all property-denoting phrases work.

The same seems true of the unbounded nature of the construction (Section 3.5). Despite working very, very hard to track down such cases, I have found only 58 PiPPs spanning finite-clause boundaries in my corpus resources (Appendix E). This seems insufficient to support the conclusion that PiPPs can span such boundaries. And none of these cases spans two or more finite-clause boundaries. Yet we all recognize such examples as grammatical.

This seems genuinely puzzling. We have no direct experience indicating that PiPPs can span multiple finite-clause boundaries, and yet we infer that such constructions are grammatical. Oh the other hand, we have no direct experience with PiPPs involving *although* as the prepositional head, and we infer that such constructions are ungrammatical. What accounts for these very different inferences? It is of course tempting to invoke very specific inductive biases of human learners, biases that cannot be learned from experience but rather are in some sense innate. This is a reasonable explanation for the above description. Before adopting it, though, we should consider whether agents that demonstrably do not have such inductive biases are able to learn to handle PiPPs. I turn to this question next.

5 Large language models

Over the last five years, large language models (LLMs) have become central to nearly all research in AI. This trend began in earnest with the ELMo model (Peters et al. 2018), which showed how large-scale training on unstructured text could lead to very rich contextualized representations of words and sentences (important precursors to ELMo include Dai and Le 2015 and McCann et al. 2017). The arrival of the Transformer architecture is the second major milestone (Vaswani et al. 2017). The Transformer is the architecture behind the GPT family of models (Radford et al. 2018, 2019, Brown et al. 2020), the BERT model (Devlin et al. 2019), and many others. These models not only reshaped AI and NLP research, but they are also having an enormous impact on society.

The Transformer architecture marks the culmination of a long journey in NLP towards models that are low-bias in the sense that they presuppose very little about how to process and represent data. In addition, when the Transformer is trained as a pure language model, it is given no supervision beyond raw strings. Rather, the model is *self-supervised*: it learns to assign high probability to attested inputs through an iterative process of making predictions at the token level, comparing those predictions to attested inputs, and updating its parameters so that it comes closer to predicting the attested strings. This can be seen as a triumph of the distributional hypotheses of Firth (1935), Harris (1954), and others: LLMs are given only information about cooccurrence, and from these patterns they are expected to learn substantive things about language.

One of the marvels of modern NLP is how much models can in fact learn about language when trained in this mode on massive quantities of text. The best present-day LLMs clearly have substantial competence in highly specific and rare constructions (Socolof et al. 2022, Mahowald 2023), novel word formation (Pinter et al. 2020, Malkin et al. 2021, Yu et al. 2020, Li et al. 2022), morphological agreement (Marvin and Linzen 2018), constituency (Futrell et al. 2019, Prasad et al. 2019, Hu et al. 2020), long-distance dependencies (Wilcox et al. 2018, 2023), negation (She et al. 2023), coreference and anaphora (Marvin and Linzen 2018, Li et al. 2021), and many other phenomena (Warstadt et al. 2019, 2020, Tenney et al. 2019, Rogers et al. 2020). The evidence for this is, at this point, absolutely compelling in my view: LLMs induce the causal structure of language from purely distributional training. They do not use language perfectly (no agents do), but they have certainly mastered many aspects of linguistic form.

5.1 Models

The primary experiments I report are with the variant of the GPT-3 model that is available via the OpenAI API as ada.[8] I chose ada because I believe it is the oldest GPT-3 variant available and thus the one most likely to match the description offered in the original GPT-3 paper (Brown et al. 2020). In particular, we know from the paper that the original GPT-3 was trained on a mixture of the Common Crawl dataset, some books datasets, and some additional Web-derived datasets (see their Section 2.2). The results of Section 4 lead me to infer that the rate of PiPPs is around 0.03% at best in these corpora as well. As the train sets for these models approach 1 trillion words, this means they might encounter 30M PiPPs – a large absolute number, but tiny relative to other phenomena and infinitesimal alongside the number of possible PiPPs.[9]

Figure 1 is a schematic diagram of GPT-3. Input sequences are represented as sequences of one-hot vectors used to look up k-dimensional vector representations in a dense embedding space for the vocabulary $\mathbf{V}$.[10] The resulting sequence of token-level vectors (the labeled gray rectangles) are the input to a series of Transformer layers. These layers are depicted as green boxes. Each green box represents a deep, complex neural network with parameters shared throughout each layer.

Attention connections are given as gray arrows. These connect the different columns of representations, and they can be seen as sophisticated ways of learning to model the distributional similarities between the different columns. GPT-3 is an autoregressive language model, meaning that it is trained to generate text left-to-right. Thus, the attention connections go back in time but not forward – future tokens have not been generated and so attending to them is impossible. The original Transformer paper (Vaswani et al. 2017) is called 'Attention is all you need' to convey the hypothesis that these very free-form attention mechanisms suffice to allow the model to learn sophisticated things about sequential data.

In the final layer of the model, the output Transformer representations are combined with the initial embedding layer to create a vector of scores over the entire vocabulary. These scores are usually given as log probabilities. In training, the output scores are compared with the one-hot encodings for the actual sequence of inputs, and the distance between these two sequences of

[8] https://platform.openai.com

[9] Do LLMs get more information about language than human babies? The standard answer is yes, but the issue is complex. Human babies encounter less language, but they encounter it as embodied creatures in complex social settings. LLMs, by contrast, experience only decontextualized snippets of text – a strange and narrow slice of the world we live in. For discussion, see Frank 2023a.

[10] For many models in this class, the token-level vectors are combined with special positional vector representations that help the model keep track of word order. I have not depicted these here.

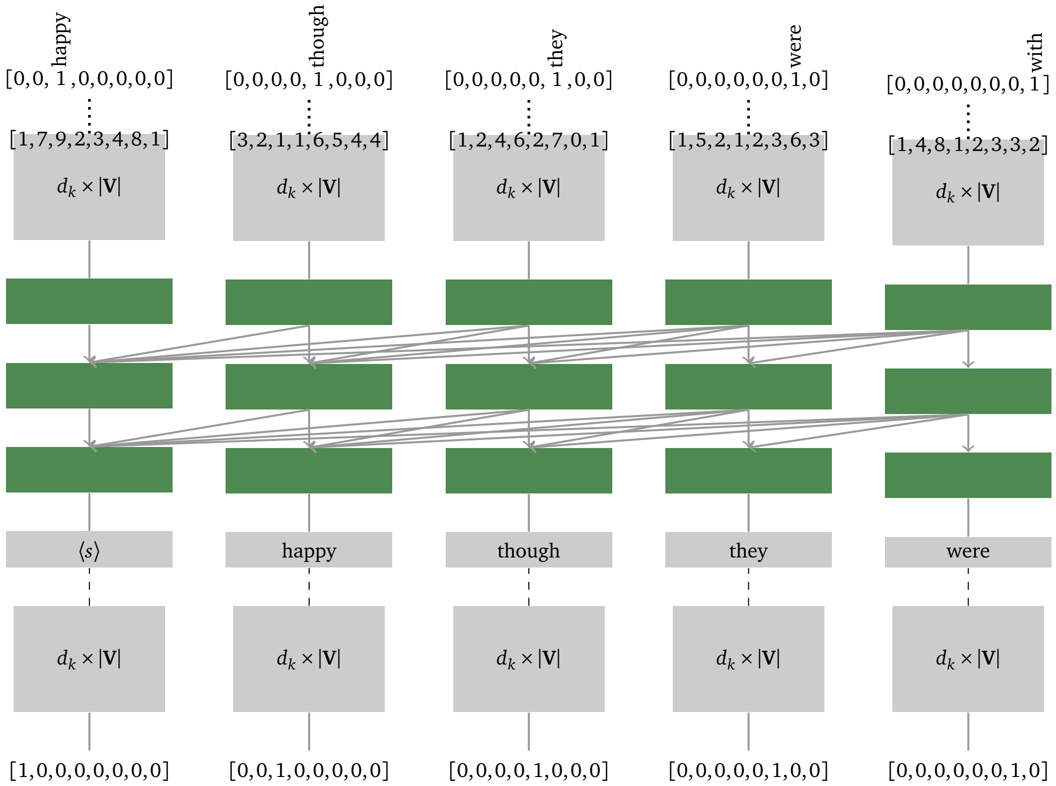


Figure 1: Schematic GPT-3 architecture diagram. This toy model has three layers and a vocab size **V** of 8. GPT-3 has 96 layers, a vocab size of around 30K items, and k (the dimensionality of almost all the model's representations) is 12,288.

vectors serves as the learning signal used to update all the model parameters via backpropagation. For our experiments, the output scores are the basis for the surprisal values that serve as our primary tool for probing models for structure. In Figure 1, the model's highest score corresponds to the actual token everywhere except where the actual token is *with*, in the final timestep. Here, the model assigns a low score to *with*, which would correspond to a high surprisal for this as the actual token. In some sense, *with* is unexpected for the model at this point. (Additional training on examples like this might change that.)

The Transformer depicted in Figure 1 has 3 layers. The GPT-3 variants used in my analyses have 96 layers. The value of k sets the dimensionality of essentially all of the representations in the Transformer. The models I use set k to 12,288. In my diagram the size of the vocabulary **V** is 8 (as seen in the dimensionality of the one-hot and score vectors). The size of the vocabulary for GPT-3 is around

50K items. This is tiny compared with the actual size of the lexicon of a language like English, because many tokens are subword tokens capturing fragments of words.[11] The entire model has 175B parameters, most of them inside the Transformer blocks.

Appendix F reports results for a few other models: GPT-2 large, GPT-3 with the `text-davinci-001` engine, and GPT-3 with the `text-davinci-003`. GPT-2 large is an older model and significantly smaller than GPT-3, at 1.54B parameters. I believe `text-davinci-001` is the version of GPT-3 used by Wilcox et al. 2023 and is similar to `ada`.

Both `ada` and `text-davinci-001` are trained with pure self-supervision. In contrast, `text-davinci-003` may be quite different, as it was additionally *Instruct fine-tuned*, meaning that it was trained on human created input–output pairs specifically designed to imbue the model with specific capabilities (Ouyang et al. 2022). OpenAI has not disclosed the nature of this Instruct data, and so for all we know their models have now been directly instructed about how PiPPs work (though this seems unlikely). For these reasons, I disfavor `text-davinci-003` (and subsequent models like ChatGPT and GPT-4) in this paper, though it should be noted that the results for all the models I tested are very similar.

The autoregressive nature of GPT-3 is limiting when it comes to studying aspects of well-formedness that might depend on the surrounding context in both directions. In particular, if we want to study whether LLMs know about the prepositional-head constraints on PiPPs (Section 3.1), the initial context will often be too short to even fully determine a PiPP parse. What we would like is to study strings like *Happy X we were with the idea*, to see what expectations the model has for *X*. Luckily, the BERT model (Devlin et al. 2019) supports exactly this kind of investigation. BERT is also based in the Transformer, but it is trained to do masked language modeling, in which the model learns to fill in missing items based on the surrounding context. The structure of BERT is schematically just like Figure 1 except the attention connections go in both directions. I use the `bert-large-cased` variant, which has 24 layers, dimensionality $k = 1{,}024$, a vocabulary of roughly 30K items, and about 340M parameters in total.

5.2 Methods

To assess whether an LLM has learned to represent PiPPs, I employ the behavioral methods of Wilcox et al. (2023): the model is prompted with examples as strings, and we compare its surprisals (i.e., negative log probabilities) at the gap site (see also Wilcox et al. 2018, Futrell et al. 2019, Hu et al. 2020).

[11] These tokenizers are also learned in a distributional fashion. GPT-3 uses the byte-pair encoding (BPE) method (Gage 1994, Sennrich et al. 2016).

In a bit more detail: as discussed above, autoregressive LLMs process input sequences token-by-token. At each timestep, they generate a sequence of scores (log probabilities) over the entire vocabulary. For instance, suppose the model processes the sequence $\langle s \rangle$ `happy though we were with`, as in Figure 1. Here, $\langle s \rangle$ is a special start token that has probability 1. The output after processing $\langle s \rangle$ will be a distribution over the vocabulary, and we can then look up what probability it assigns to the next token, `happy`. Similarly, when we get all the way to `were`, we can see what probability the model assigns to the token `with` as the next token. The surprisal is the negative of the log of this probability value. Lower surprisal indicates that the token `with` is more expected by the model. In Figure 1, `with` has low log probability, i.e., high surprisal.

Wilcox et al. (2023) use surprisals to help determine whether models know about filler–gap dependencies, using sets of items like the following:[12]

(31) a. I know what the lion devoured ____ yesterday. (Filler/Gap)

b. *I know that the lion devoured ____ yesterday. (No Filler/Gap)

(32) a. I know that the lion devoured the gazelle yesterday. (No Filler/No Gap)

b. *I know what the lion devoured the gazelle yesterday. (Filler/No Gap)

The examples in (31) contain gaps. For these, Wilcox et al. (2023) define the *wh-effect* as the difference in surprisal for the post-gap word *yesterday* between the long-distance dependency case (31a) and the minimal variant without that dependency (31b):

(33) $$-\log_2 P(\texttt{yesterday} \mid \texttt{I know what the lion devoured}) - \\ -\log_2 P(\texttt{yesterday} \mid \texttt{I know that the lion devoured})$$

In the context of an autoregressive neural language model like GPT-3, the predicted scores provide these conditional probabilities. We expect these to be large negative values, since the left term will have low surprisal and the right term will be very surprising indeed.

We can perform a similar comparison between the cases without gaps in (32):

(34) $$-\log_2 P(\texttt{the} \mid \texttt{I know that the lion devoured}) - \\ -\log_2 P(\texttt{the} \mid \texttt{I know what the lion devoured})$$

[12] It's assumed here that *devour* is obligatorily transitive. Glass (2021) shows that intransitive uses of *devour* are possible and often motivated by specific contextual factors. This doesn't challenge the method, as we require only that (31b) be high surprisal given the context provided.

For these comparisons, we expect positive values: *the* is a highly expected element in the lefthand context and unexpected in the righthand context. An important caveat here is that the gap in the filler–gap dependency could be later in the string (as in *I know what the lion devoured the gazelle with*), and so the positive values here are expected to be modestly sized.

I would also like to probe models for the prepositional-head limitations discussed in Section 3.1. However, we can't simply extend the wh-effects idea to these phenomena, for two reasons. First, we need to compare different lexical items, whereas the above hypotheses assess the same item conditional on different contexts. Second, the prepositional head in a PiPP occurs too early in the construction to ensure that the PiPP parse is even a dominant one for a model (or any agent processing the input in a temporal order).

To address these issues, I propose to use the BERT model to process examples. As discussed above, BERT uses bidirectional context, so we can ask it for the score of a word that we have masked out in the entire string. To address the concern that different words will have different prior probabilities, I propose to compare the PiPP construction with its minimal grammatical variant, the regular PP construction. Bringing these ideas together, this means that we consider pairs of examples like the following:

(35) a. `[MASK] they were tired, they pressed on.` (PiPP)

b. `Tired [MASK] they were, they pressed on.` (PP)

At these `[MASK]` sites, BERT predicts a distribution of scores over the entire vocabulary, just as GPT-3 does. Here, though, the scores are influenced by the entire surrounding context. For a given preposition P, we compare the surprisal for P in the PiPP with the surprisal in the PP. The difference is the *prepositional-head effect* for PiPP.

5.3 Materials

Wilcox et al. (2023) show that both wh-effects (33) and (34) are robustly attested for GPT-3 as well as a range of smaller models. Their methodology is easily adapted to other unbounded dependency constructions, and so we can ask whether similar effects are seen for PiPPs. To address this question, I created a dataset of 33 basic examples covering a range of different predicators, preposed phrases, and surrounding syntactic contexts. Each of these sentences can be transformed into four items reflecting the four conditions we need in order to assess wh-effects.

Item	Condition	Prep.	Embedding
Happy though we were **with** the idea, we had to reject it.	Filler/Gap (PiPP)	though	None
*Though we were **with** the idea, we had to reject it.	No Filler/Gap	though	None
*Happy though we were **happy** with the idea, we had to reject it	Filler/No Gap	though	None
Though we were **happy** with the idea, we had to reject it.	No Filler/No Gap	though	None

Table 1: Sample experimental item. To obtain variants with Prep *as* or *although*, we change *though* and capitalize as appropriate. To create embedding variants, we insert the fixed string *they said that we knew that* right before the PiPP prepositional head. The target word is in bold. This is the word whose surprisal we primarily measure.

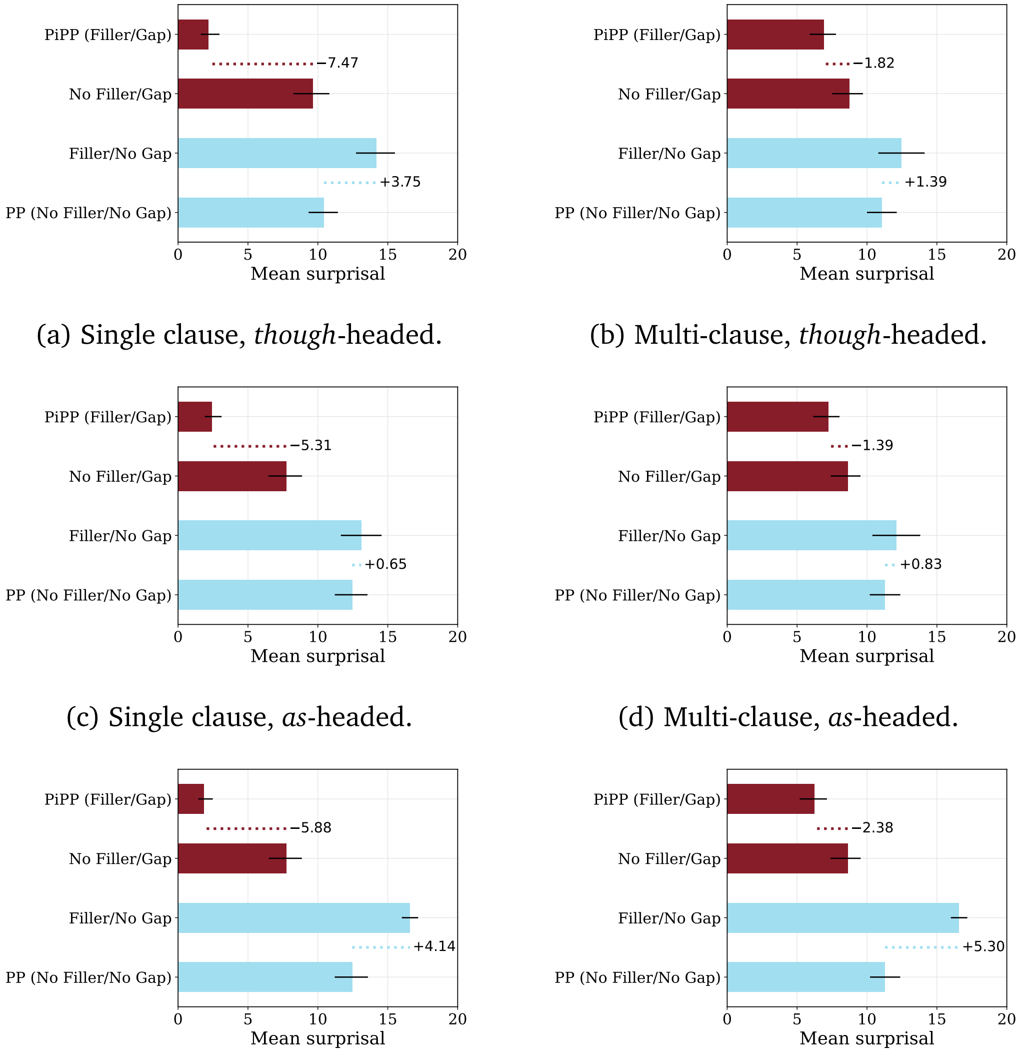
An example of this paradigm is given in Table 1. Each item can be automatically transformed into ones with different prepositional heads, and we can add embedding layers by inserting strings like *they said that* directly after the PiPP head preposition. I consider three head-types in this paper: *as*, *though*, and *as*. . . *as*. The final variant is not strictly speaking a variant in terms of the prepositional head, but it is the most common type in my corpus studies and so it seems useful to single it out for study rather than collapsing it with the less frequent plain *as* variants.

The materials for these experiments are included in the code repository for this paper: https://github.com/cgpotts/pipps.

5.4 Results

Figure 2 summarizes the results for the ada engine. Each pair of panels shows a different prepositional head. The single-clause items are on the left and multi-clause items are on the right. The multi-clause variants are created using the fixed string *they said that we knew that*, which results in PiPPs that span two finite-clause boundaries.

The dotted lines indicate the two wh-effects. As noted above, we expect the wh-effect for the gap cases (red bars) to be large and negative, and the wh-effects for the gapless cases (blue bars) to be positive and modest in size.



(a) Single clause, *though*-headed.

(b) Multi-clause, *though*-headed.

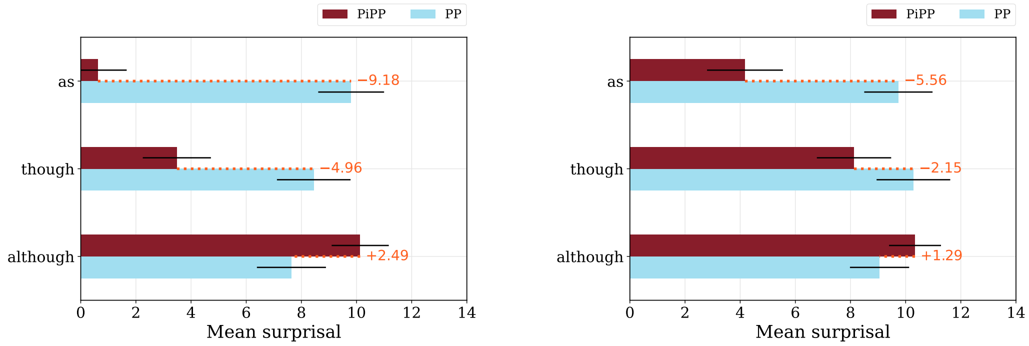
(c) Single clause, *as*-headed.

(d) Multi-clause, *as*-headed.

(e) Single clause, *as. . . as*-headed.

(f) Multi-clause, *as. . . as*-headed.

Figure 2: Core wh-effects for the GPT-3 ada model. The bars represent means surprisals for the target word, with bootstrapped 95% confidence intervals. The wh-effects are indicated with dotted lines, and the numerical value of those effects is given (based on the surprisal means).



(a) Single clause.

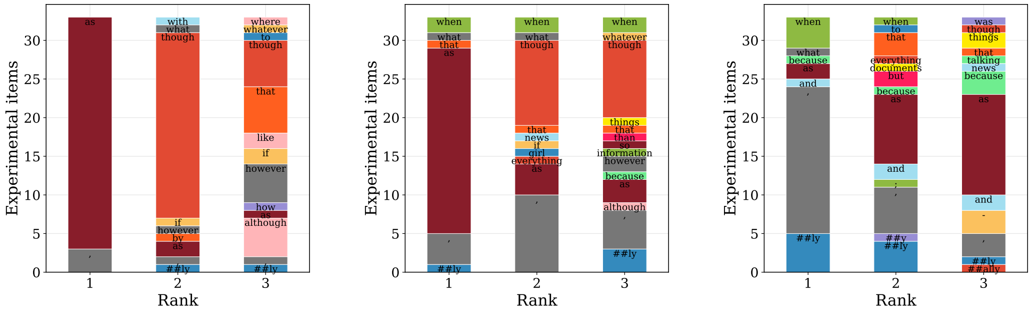
(b) Multi-clause.

Figure 3: Prepositional-head comparisons using BERT.

In all contexts, the expected wh-effects emerge. These same patterns hold for the other models I tested (Appendix F), though the later GPT-3 variants seem to consistently reverse the expected wh-effect for multi-clause *as*-headed cases. We expect these effects to be small. The more important ones for our purposes are those that compare gap-containing cases. The wh-effects for those are strong (strongly negative, as hypothesized) for all models.

Figure 3 summarizes the findings for the prepositional-head effect, for single clause and multi-clauses cases. For these experiments, I used the same materials as for the filler–gap experiments, but the model is now BERT, as discussed above. It seems clear that BERT finds *although* extremely surprising in PiPPs. Strikingly, on average, *although* is the lowest surprisal of the prepositions tested in the regular PP cases like (35b). The PiPP context reverses this preference. In contrast, *though* and *as* are low surprisal in PiPP contexts as compared to the PP context.

We can probe deeper here. The prepositional-head constraints lead us to expect that *though* and *as* will be the top-ranked choices for PiPPs. Figure 4 assesses this by keeping track of which words are top-scoring in each of the 33 items, for the `[MASK]` position corresponding to the prepositional head. For the single-clause cases (Figure 4a), *as* is the top prediction for 30 of the 33 items, and *though* is the second-place prediction for 25 of the 33 items. This looks like an almost categorial preference for these items. Interestingly, when we insert a single finite-clause boundary (Figure 4b), these preferences are less clear, though *as* and *though* remain dominant. For the double embedding (Figure 4c), the preference for *as* and *though* has mostly disappeared. This is interesting when set along side the clear gap-sensitivity for these multi-clause embeddings in Figure 2



(a) Single clause.

(b) Embedding with *they said that*.

(c) Embedding with *they said that we knew that*.

Figure 4: Ranking of PiPP prepositional heads for BERT, at different levels of embedding.

(though those results are for GPT-3 and these are for BERT, so direct comparisons are speculative).

5.5 Discussion

GPT-3 seems to have learned to latently represent PiPPs at least insofar as it has an expectation that (1) a PiPP gap will appear conditional on an earlier PiPP filler configuration, and (2) the prepositional head will be *as* or *though*. These expectations hold not only in the single-clauses case but also in the sort of multi-clause context that we know to be vanishingly rare even in massive corpora like those used to train the models (Figure 2).

Where does this capacity to recognize PiPPs come from? In thinking about humans, it was reasonable to imagine that specific inductive biases might be at work in allowing the relevant abstract concepts to be learned. For LLMs, this is not an option: the learning mechanisms are very general and completely known to us, and thus any such inductive biases must not be necessary. This does not rule out that the human solution is very different, but it shows that the argument for innate learning mechanisms will need to be made in a different way. There evidently is enough information in the input strings for the learning task at hand.

The above experiments are just the start of what could be done to fully characterize what LLMs have learned about PiPPs. We could also consider probing the internal representations of LLMs to assess whether they are encoding more abstract PiPP features directly. For example, we might ask whether a preposed phrase followed by a PiPP preposition triggers the model to begin tracking that it is in an unbounded dependency state. Ravfogel et al. (2021) begin to develop such methods for relative clause structures. More recent intervention-based methods for model explainability seem ideally suited to these tasks (Geiger et al. 2021, 2022, 2023a,b, Wu et al. 2023). We could process minimal pairs like those used in our experiments, swap parts of their internal Tranformer representations, and see whether this has a predictable effect on their expectations with regard to gaps. This would allow us to identify where these features are stored in the network. These experiments would take us beyond behavioral testing to more deeply understand the latent structures models have learned.

One final note: one might wonder whether LLMs can perform the intuitive transformation that relates PPs to PiPPs, as in *Though we were happy* ⇒ *Happy though we were*. I should emphasize that I absolutely do not think this ability is a prerequisite for being proficient with PiPPs. Many regular human uers of PiPPs would be unable to perform this transformation in the general case. Still, the question of whether LLMs can do it is irresistible. I take up the question in Appendix G. The quick summary: LLMs are good at this transformation.

6 Model-theoretic syntax characterization of PiPPs

It seems that both people and LLMs are able to become proficient with PiPPs despite very little experience with them. Moreover, this proficiency entails a few different kinds of inference from data: for some properties (prepositional heads, dropping the determiner in preposed nominals), the learner needs to infer that the attested cases exhaust the possibilities. For other properties (which phrases can be preposed, where gaps can occur), the inferences need to generalize beyond what exposure would seem to support. In addition, the LLM evidence suggests that a simple, uniform learning mechanism suffices to achieve this. What sort of theoretical account can serve as a basis for explaining these observations?

In this section, I argue that model-theoretic syntax (MTS) is an excellent tool for this job. In MTS, grammars take the form of collections of constraints on forms. More precisely, we cast these constraints as necessary (but perhaps not sufficient) conditions for well-formedness by saying that a form is licensed only if it satisfies all the constraints. Rogers (1997, 1998) showed how to define prominent generative approaches to syntax in MTS terms and began to identify the consequences of this new perspective. Pullum and Scholz (2001) trace the history of the ideas and offer a visionary statement of how MTS can be used both to offer precise grammatical descriptions and to address some of the foundational challenges facing generative syntactic approaches in general. Pullum (2007a, 2020) refines and expands this vision.

In offering an MTS description of PiPPs, I hope to further elucidate the nature of the construction. However, I seek in addition to connect the MTS formalism with the very simple learning mechanisms employed by LLMs. In essence, this reduces to the scores that LLMs assign to the vocabulary at each timestep. In training, these scores are continually refined to be closer to the vectors for the training sequences. In this way, frequent patterns achieve higher scores, and infrequent patterns get low scores. What counts as a "pattern" in this context? That is a difficult question. We know from the results I summarized at the start of Section 5, and from our lived experiences with the models themselves, that they are able to identify extremely abstract patterns that allow them to recognize novel sequences and produce novel grammatical sequences.

My MTS description will be somewhat informal to avoid notational overload. The constraints themselves all seem to be of a familiar form, and it is hard to imagine a reader coming away from reading Rogers (1998) or Pullum and Scholz (2001) with concerns that MTS grammars cannot be made formally precise, so I think an informal approach suffices given my current goals.

6.1 Gap licensing

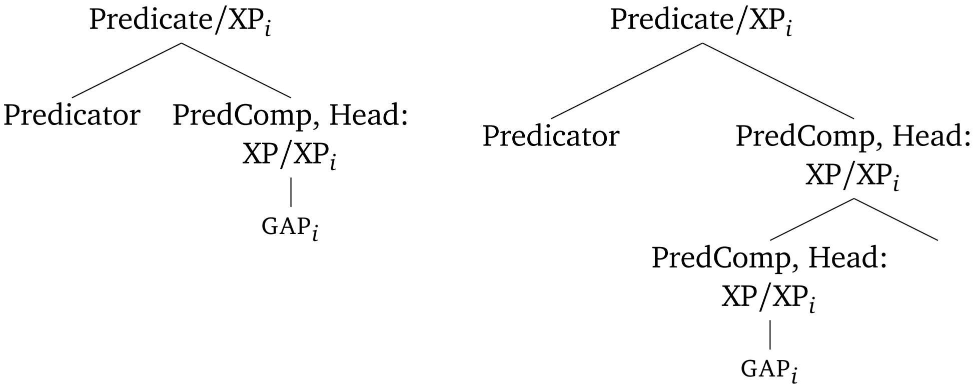
Let's begin with the most substantive and interesting constraint on PiPPs: the gap licensing environment. The following states the proposed constraint:

(36) If a node N has category XP and a child with the feature GAP$_i$, then N has the features PredComp and /XP$_i$ (for some variable i).

Here, XP is a variable over phrasal syntactic categories. I assume that the feature PredComp is itself licensed on a node only if that node is the complement of a Predicator node or part of a head path that ends in such a complement node.

Constraint (36) centers on the gap site, enforcing requirements for the surrounding context. The goal is to license gaps in the following sort of configurations:

(37)



On the left, we have the simple case where the relevant PredComp is the direct complement of a Predicator. On the right, we have a head path of two nodes. This opens the door to the sort of modifier stranding we saw in Section 3.4.

The above constraint does not cover PiPPs in which the preposed phrase is an adverb, as in (16). For example (16b), there is a case to be made that the adverb is a complement of the predicate, but this seems less plausible for (16a). I leave accounting for these cases as a challenge for future work.

The complex feature $\mathrm{XP}/\mathrm{XP}_i$ begins to track the filler–gap dependency, in essentially the same manner as is done in GPSG (Gazdar et al. 1985). In (37), I have shown how this would be inherited through the chain of nodes that constitute the head path for the PredComp and up to the Predicate node. The full MTS grammar should include constraints that manage the series of local dependencies that make up these unbounded dependencies constructions. Such an MTS theory is given in full for both GPSG and GB in Rogers 1998.

Arguably the most important feature of (36) is that it does not have any PiPP-specific aspects to it. Any predicational environment of the relevant sort is expected to license gaps in this way, all else being equal. This seems broadly correct, as PiPPs are just one of a number of constructions that seem to involve this same local structure:

(38) a. They are happier than we are.
b. They are as happy as we are.
c. ^{B}Poor as church mice they were, but it didn't matter.
d. They wanted to run the race, and run the race they did.
e. ^{B}the view, such as it was, never failed to intimate that reality is negligible as dreams
f. ^{C}... however amusing the posturing and gestures may seem it is in extremely bad taste to laugh, make asides etc and it will give deep offence - it is not a case where the customer is always right.

The comparative construction in (38a) and (38b) may be the key to all of this: such constructions are incredibly common. If learners are able to infer from them that they contain gaps and those gaps are licensed by predicators – that is, if they infer the latent structure depicted in (37) – then they have learned a substantial amount about PiPPs even if they never encounter an actual PiPP.

6.2 Prenucleus constraints

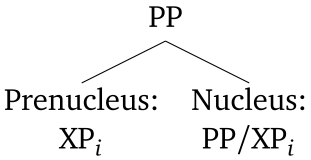
Constraint (36) licenses an unbounded dependency gap element, and we assume that this dependency is passed up through a series of local feature relationships.

The following constraint requires that this dependency be discharged at the top of the PiPP construction:

(39) If a PP node *N* has child nodes labeled Prenucleus and Nucleus, then the Prenucleus has feature XP_i (for some variable *i*), the Nucleus has feature $\mathrm{PP}/\mathrm{XP}_i$, neither of them has any other slash features, and *N* does not have any slash features.

This describes trees like (40) and entails that the PiPP unbounded dependency is discharged here.

(40)



In addition, it entails that PiPPs can contain only one unbounded dependency and that they are islands for unbounded dependencies.

We could supplement (39) with additional constraints on the Prenucleus phrase, for example, to block determiners (Section 3.3.2) and to allow adverbial *as* (Section 3.3.1).

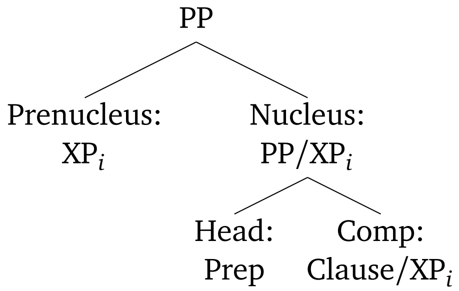
Importantly, nothing about the above set of constraints requires that the Prenucleus element would be grammatical if placed in the gap site. There is no "movement" in any formal sense. The constraints center around the dependency, which tracks only an index and a syntactic category type.

This constraint is very close to being PiPP-specific; the local tree it describes (40) is certainly indicative of a PiPP. It may be fruitful to generalize it to cover the way slash dependencies are discharged in the constructions represented in (38) and perhaps others.

6.3 Prepositional head constraints

The final constraint I consider is the prepositional-head constraint. It is highly specific to PiPPs:

(41) If T matches the form,



then the child node of the Head:Prep node in T is *though* or *as*.

For LLMs, this be reflected in fact that they will assign very low scores to other prepositions in this environment. People may do something similar and intuitively feel that those low scores mean the structures are ungrammatical.

A fuller account would refer to the semantics of the prepositional head, in particular, to specify that if *as* has a concessive reading, then it is in the above environment.

Why is constraint (41) so much more specific than the others we have given so far? There may not be a deep answer to this question. After all, it is easy to imagine a version of English in which PiPP licensing is broader. On the other hand, many constructions are tightly associated with specific prepositions, so LLMs (and people) may form a statistical expectation that encounters with prepositions should not be generalized to other forms in that class.

6.4 Discussion

I offered three core constraints: one highly PiPP-specific one relating to prepositional heads (41), one that mixes PiPP-specific things with general logic relating to discharging unbounded dependencies (39), and one that is general to gap licensing (36). Taken together, these capture the core syntactic features of PiPPs.

It seems natural to infer from this description that PiPPs are, in some sense, epiphenomenal – the consequence of more basic constraints in the grammar. From this perspective, we might not be able to clearly and confidently say exactly which constructions do or don't count as PiPPs. For example, the adverbial cases in (16) might be in a gray area in terms of PiPP status. But 'PiPP" is a post hoc label without any particular theoretical status, and so lack of clarity about its precise meaning doesn't mean that the theory is unclear.

I am confident that the constraints can be learned purely from data by sophisticated LLMs. For the prepositional-head constraint, this seems like a straightforward consequence of LLM scoring. For the other constraints, we need to posit that LLMs induce latent variables for more abstract features relating to syntactic categories, constituents, and slash categories. The precise way this happens remains somewhat mysterious, but I cited extensive experimental evidence that it does arise even in LLMs trained only with self-supervision on unstructured text.[13] The final state that LLMs are in after all this will also not reify PiPPs as a specific construction. Rather, PiPPs will arise when the model's inputs and internal representations are in a particular kind of state, and this will be reflected in how they score both well-formed PiPPs and ill-formed ones, as we saw in Section 5.

7 Conclusion

The origins of this paper stretch back to a challenge Geoff Pullum issued in the year 2000: find some naturally occurring PiPPs spanning finite-clause boundaries. With the current paper, I feel I have risen to the challenge: conducting numerous highly motivated searches in corpora totaling over 7.6B sentences, I managed to find 58 cases (see Section 3.5 and Appendix E).

This paper was partly an excuse to find and present these examples to Geoff. However, I hope to have accomplished more than that. The massive corpora we have today allowed me to further support the *CGEL* description of PiPPs, and perhaps modestly refine that description as well (Section 3). We can also begin to quantify the intuition that PiPPs are very rare in usage data. Section 4 estimates that around 0.03% of sentences contain them, compared to 12% for restrictive relative clauses (a common unbounded dependency construction).

The low frequency of PiPPs raises the question of how people become proficient with them. It is tempting to posit innate learning mechanisms that give people a head start. Such mechanisms may be at work, but data sparsity alone will not carry this argument: I showed in Section 5 that present-day LLMs are also excellent PiPP recognizers. Their training data also seem to underdetermine the full nature of PiPPs, and yet LLMs learn them. This suggests an alternative explanation on which very abstract information is shared across different con-

[13] Bhattacharya and van Schijndel (2020), Mitchell and Bowers (2020) and Lasri et al. (2022) suggest that earlier LLMs learn in a more fragmentary way, with minimal sharing of information across related constituents. This may also be true of GPT-3, but it seems likely that LLMs will continue to improve in this regard.

texts, so that PiPPs emerge from more basic elements rather than being acquired from scratch. I offered an MTS account that I think could serve as a formal basis for such a theory of PiPPs and how they are acquired.

Geoff's research guided me at every step of this journey: the initial PiPP challenge, the *CGEL* description, the role of corpus evidence, the nature of stimulus poverty arguments, and the value of MTS as a tool for formal descriptions that can serve a variety of empirical and analytical goals. What is next? Well, Geoff already implicitly issued a follow-up challenge when commenting on Mark Davies' example (6b):

> That is enough to settle my question about whether the construction can have an unbounded dependency, provided we assume – a big but familiar syntactician's assumption – that if the gap can be embedded in one finite subordinate clause it can be further embedded without limit. (Pullum 2017: 290).

A clear, careful generalization from data, and a clear statement of the risk that the generalization entails. To reduce the risk, we need at least one naturally occurring PiPP case spanning at least two finite-clause boundaries. On the account I developed here, such an example would provide no new information to linguists or to language users, but it still felt important to me to find some. With the help of a powerful NLP model (Appendix D) and some intricate regexs, I searched through the roughly 7.6B sentences in C4 and BookCorpusOpen, and I eventually found three double finite-clause cases:

(42) a. $^{\mathrm{C}}$Well-intentioned though many people may have imagined that the CIA probably thought they were, their foreign-policy operations were confused, duplicitous failures.[14]

b. $^{\mathrm{C}}$As for planning, as sinister as I think this student thinks our meetings may be, they are really not!

c. $^{\mathrm{C}}$As much of a downer as I think we both agree the pistols are, for us, do you not find the only thing worse than using one yourself is when someone else in the lobby absolutely dominates with them, when running them akimbo?

[14] This example is from the English Language Learners Stack Exchange. The user is asking for help in understanding the PiPP. Another user offers the regular PP as an explanation. I have not been able to find the original source of the example sentence.

References

Jack Bandy and Nicholas Vincent. 2021. Addressing "documentation debt" in machine learning research: A retrospective Datasheet for BookCorpus. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*. Curran.

Debasmita Bhattacharya and Marten van Schijndel. 2020. Filler-gaps that neural networks fail to generalize. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 486–495, Online. Association for Computational Linguistics.

T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, P. Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, Tom Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Mark Davies. 2008. The Corpus of Contemporary American English. Available online at http://corpus.byu.edu/coca/.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, and Matt Gardner. 2021. Documenting the english colossal clean crawled corpus. *ArXiv*, abs/2104.08758.

Emmanuel Dupoux. 2018. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59.

John R. Firth. 1935. The technique of semantics. *Transactions of the Philological Society*, 34(1):36–73.

Michael C. Frank. 2023a. Bridging the data gap between children and large language models. PsyArXiv.

Michael C Frank. 2023b. Large language models as models of human cognition. *PsyArXiv*.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Gerald Gazdar, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge: Harvard University Press and London: Basil Blackwell.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumeé III, and Kate Crawford. 2018. Datasheets for datasets. *arXiv preprint arXiv:1803.09010*.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586.

Atticus Geiger, Christopher Potts, and Thomas Icard. 2023a. Causal abstraction for faithful model interpretation. Ms., Stanford University.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. 2022. Inducing causal structure for interpretable neural networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 7324–7338. PMLR.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. 2023b. Finding alignments between interpretable causal variables and distributed neural representations. Ms., Stanford University.

Jill Gilkerson, Jeffrey A. Richards, Steven F. Warren, Judith K. Montgomery, Charles R. Greenwood, D. Kimbrough Oller, John H. L. Hansen, and Terrance D. Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

Lelia Glass. 2021. English verbs can omit their objects when they describe routines. *English Language and Linguistics*, pages 1–25.

Zellig Harris. 1954. Distributional structure. *Word*, 10(23):146–162.

Betty Hart and Todd R. Risley. 1995. *Meaningful Differences in the Everyday Experience of Young American Children*. Paul H. Brookes, Baltimore, MD.

Philip Hofmeister and Ivan A. Sag. 2010. Cognitive constraints and island effects. *Language*, 22(6):366–415.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Rodney Huddleston and Geoffrey K. Pullum. 2002. *The Cambridge Grammar of the English Language*. Cambridge University Press, Cambridge.

Christopher Kennedy. 2007. Vagueness and grammar: The semantics of relative and absolute gradable adjective. *Linguistics and Philosophy*, 30(1):1–45.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational linguistics*, 32(4):485–525.

Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. 2022. Probing for the usage of grammatical number. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8818–8831, Dublin, Ireland. Association for Computational Linguistics.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. 2021. Implicit representations of meaning in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1813–1827, Online. Association for Computational Linguistics.

Siyan Li, Riley Carlson, and Christopher Potts. 2022. Systematicity in GPT-3's interpretation of novel English noun compounds. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 717–728, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. ArXiv:1907.11692.

Kyle Mahowald. 2023. A discerning several thousand judgments: GPT-3 rates the article + adjective + numeral + noun construction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 265–273, Dubrovnik, Croatia. Association for Computational Linguistics.

Nikolay Malkin, Sameera Lanka, Pranav Goel, Sudha Rao, and Nebojsa Jojic. 2021. GPT perdetry test: Generating new meanings for new words. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5542–5553, Online. Association for Computational Linguistics.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural*

Language Processing, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6294–6305. Curran Associates, Inc.

Jeff Mitchell and Jeffrey Bowers. 2020. Priorless recurrent networks learn curiously. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5147–5158, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Steven Piantadosi. 2023. Modern language models refute Chomsky's approach to language. *Lingbuzz Preprint, lingbuzz*, 7180.

Yuval Pinter, Cassandra L. Jacobs, and Jacob Eisenstein. 2020. Will it unblend? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1525–1535, Online. Association for Computational Linguistics.

Paul M. Postal. 1998. *Three Investigations of Extraction*. Cambridge, MA, MIT Press.

Grusha Prasad, Marten van Schijndel, and Tal Linzen. 2019. Using priming to uncover the organization of syntactic representations in neural language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Geoffrey K Pullum. 2007a. The evolution of model-theoretic frameworks in linguistics. In *Model-Theoretic Syntax at 10*, volume 10, pages 1–10.

Geoffrey K. Pullum. 2007b. Ungrammaticality, rarity, and corpus use. *Corpus Linguistics and Linguistic Theory*, 3(1):33–47.

Geoffrey K. Pullum. 2009. Computational linguistics and generative linguistics: The triumph of hope over experience. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics:*

Virtuous, Vicious or Vacuous?, pages 12–21, Athens, Greece. Association for Computational Linguistics.

Geoffrey K. Pullum. 2017. Theory, data, and the epistemology of syntax. In Angelika Konopka, Marek und Wöllstein, editor, *Grammatische Variation. Empirische Zugänge und theoretische Modellierung*, pages 283–298. de Gruyter.

Geoffrey K. Pullum. 2020. Theorizing about the syntax of human language: A radical alternative to generative formalisms. *Cadernos de Linguística*, 1(1):1–33.

Geoffrey K. Pullum and Barbara C. Scholz. 2001. On the distinction between model-theoretic and generative-enumerative syntactic frameworks. In Philippe de Groote, Glyn Morrill, and Christian Retoré, editors, *Logical Aspects of Computational Linguistics: 4th International Conference, LACL 2001*, pages 17–43. Springer, Berlin.

Geoffrey K. Pullum and Barbara C. Scholz. 2002. Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, 19(1-2):9–50.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Ms, OpenAI.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 1–67.

Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. 2021. Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 194–209, Online. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. ArXiv:2002.12327.

James Rogers. 1997. “Grammarless” phrase structure grammar. *Linguistics and Philosophy*, 20(6):721–746.

James Rogers. 1998. *A Descriptive Approach to Language-Theoretic Complexity*. CSLI/FoLLI, Stanford, CA.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jingyuan Selena She, Christopher Potts, Samuel R. Bowman, and Atticus Geiger. 2023. ScoNe: Benchmarking negation reasoning in language models with fine-tuning and in-context learning. To appear in Proceedings of ACL.

Michaela Socolof, Jackie Cheung, Michael Wagner, and Timothy O'Donnell. 2022. Characterizing idioms: Conventionality and contingency. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4024–4037, Dublin, Ireland. Association for Computational Linguistics.

Daniel Sullivan. 2005. Search engine sizes. Search Engine Watch.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alex Warstadt and Samuel R Bowman. 2022. What artificial neural networks can tell us about human language acquisition. *Algebraic Structures in Natural Language*, pages 17–60.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Ethan Gotlieb Wilcox, Richard Futrell, and Roger Levy. 2023. Using computational models to test syntactic learnability. *Linguistic Inquiry*, pages 1–44.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D. Goodman. 2023. Interpretability at scale: Identifying causal mechanisms in Alpaca. Ms., Stanford University.

Zhiwei Yu, Hongyu Zang, and Xiaojun Wan. 2020. Homophonic pun generation with lexically constrained rewriting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2870–2876, Online. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

Supplementary materials

A My message to Geoff

```
From potts@ling.ucsc.edu Tue Dec 24 05:33:49 2002
Date: Tue, 24 Dec 2002 05:33:48 -0800 (PST)
From: Christopher Potts <potts@ling.ucsc.edu>
To: Geoff Pullum <pullum@ling.ucsc.edu>
Subject: Long-distance left extern

Geoff!

  Although he sometimes retreated to a stance of pure practicality,
 Feynman gave answers to these questions, philosophical and unscientific
  though he knew they were.
 ---James Gleick. 1992. Genius: The Life and Science of Richard Feynman.
     New York: Vintage Books (p. 13).

In addition to sporting this rarity, it's a terrific biography.  I have
an extra copy if you'd like it for the trip back from Atlanta.  I read it
many years ago, many years before I knew to look for long distance fronted
PP left externs.

---Chris
```

B Geoff’s message

```
From: "Geoffrey K. Pullum" <gpullum@ling.ed.ac.uk>
Subject: a case from speech
Date: April 12, 2011 at 00:55:05 PDT
To: cgpotts@stanford.edu

Just a minute ago I heard someone speaking on BBC Radio 4 (not reading from
a script) say something about the European Court of Human Rights that began:

    Unpopular though I can well see that it might be, ...

At last, confirmation of the unboundedness from speech!  The predicate
preposing of the "happy though I am" PP construction is, indeed, an
unbounded dependency.
```

```
I've been looking for good attested examples of that sort for about ten
years, as you well know.  You found the first one, in Gleick's biography
of Feynman, when you were a puppy.  But from spontaneous speech!  This is
a red letter day for evidence-based linguistics.

Best wishes,
GKP
```

C Initial regular expression

The following is the regex I used to create initial samples to annotate (see Section 4.2, step 1):

```python
import re
main_regex = re.compile(r"""
    (\S+)
    \s+
    (?:though|as)
    \s+
    (?:\S+\s+)+""", re.VERBOSE | re.I)
```

I add an additional step of filtering off examples where the case-normalized initial matching group is in the set {as, even, but, and}. As far as I can tell, the only risk this runs is in filtering off cases like *Even though the odds were, we still lost*. This seems preferable to ending up with samples that are totally dominated by phrases headed with *even though*.

D PiPP classifier

I ended up annotating and collecting a lot of examples as part of doing the work for Sections 3–4. At a certain point, I realized I had collected enough examples that it seemed plausible that I would be able to train a classifier to help me find more useful examples. This turned out to be extremely productive.

To build this classifier, I began with `BERT-base-cased` parameters. I had 7,043 annotated examples: 6,598 negative cases and 445 positive cases. To get a sense for how effective these models could be, I first split this dataset randomly into 80% train and 20% test examples, which resulted in a test set containing 1,314 negative cases and 95 positive cases – very close to the overall distribution. I trained the model using a HuggingFace `transformers` training protocol (Wolf et al. 2019), with the only departure from the defaults being that I assigned the positive class twice the weight of the negative class to help make up for the class imbalance. I trained the model using a Google Colab notebook with GPU support. This took about 15 minutes.

This model achieved an F1 of 0.99 for the negative class and 0.92 for the positive class, for a macro F1 of 0.96. This suggested to me that this classifier approach could be successful, so I retrained the model on all my labeled examples using the same protocol. As a check, I evaluated this model on every PiPP and PP case from my Section 5.3 materials, in every prepositional head and embedding variant. This led to a global accuracy of 0.99 (two mistakes in the entire test). On the stress-test case used in Appendix G, the model also got 17 of 18 examples correct. Overall, the results struck me as excellent, so I began using the model to find more examples.

The code for the model is included in the project repository, and the model is available on the HuggingFace Model Hub as `cgpotts/pipp-finder-bert-base-cased`. This may be the most obscure and specific model on the HuggingFace Model Hub right now.

E Naturally occurring PiPPs spanning finite clause boundaries

1. [B] Honourable though I am sure his intentions were, he betrayed you, Ruben.
2. [B] Hold thy explanation, excellent though I'm sure it is!
3. [B] "they 're right here," she told him and, unlikely though she knew it was, she couldn't help wishing he'd squeeze her hand – or give her some other small token on which to hang all her hope.
4. [B] Eriks reassurance, heart-felt though she knew it was, did little to ease her anxiety over the impending day.
5. [B] Impossible as she thought it would be for anyone else, she swallowed the cold coffee and began once again to type.
6. [B] Celibate as he wished they were
7. [B] Weak though I thought they were, when she stamped closer the ground shook beneath us.
8. [B] Strange as you may think it is, the soup shortage was part of one ...
9. [B] Busy as you say you are, I thought you'd be happy I saved you the extra work.
10. [B] Guilty though I believe Mars to be

11. [B]Prepared as he thought he was for this confrontation, his knees still buckled in anticipation for what Antone said next.

12. [B]Oddly, as wrong as she knew it was for Noah to assume his behavior was acceptable, a very small part of her was thrilled by it.

13. [C]We were agog for the Memories of Max Miller and his Life in the Theatre, risqué though we feared the Cheeky Chappie might be, from the peerless Mr Bill Pertwee, who spared not our blushes and who appeared in a Dressing Gown of sorts that almost beggared belief, even though redeemed in full by the familiar hat & patter.

14. [C]I will remember that the collective wisdom of gardening and knowledge of plants is much bigger than my knowledge of gardening, vast though I think it might be, and therefore, it is possible for a new gardener to encounter some new wisdom or knowledge that I know nothing about.

15. [C]That, for me, is what the whole thing is about, anachronistic though I fear it may be.

16. [C]They are many and, complex though I like to think I am, I am not legion.

17. [C]In my judgment the Secretary of Statement made it quite sufficiently clear that what she was doing was simply reaching a different judgment about the degree of harm, significant though she agreed that it was.

18. [C]Amen to that, optimistic though I fear it may be.

19. [C]He was trying to make a noise; to ward something off or drown something outwhat, I could not imagine, awesome though I felt it must be.

20. [C]You must understand how embarassing it was to discover this melon felony, inadvertent though I assure you it was!

21. [C]We reached for our cameras, inadequate though we knew they would be.

22. [C]I am hoping that there will be a second referendum, tedious though I know this will be.

23. [C]So, foolish though I think you are, marginal though I know you to be, you people do indeed manage to do far more real damage than any sensible person would suspect on a superficial looksee.

24. [C]However, early though we thought our arrival at the gate was, many had thought to come much earlier and the line was already halfway up the Mall.

25. ^{C}You are not like him, Elijah, as much as you seem to think you are.

26. ^{C}As inept as Giovanni and Carwyn seemed to think Lorenzo was about technology, why did he have a financial guy who had online access in his super-secret bad guy lair?

27. ^{C}Prepared as we think we are, we’re defeated.

28. ^{C}Beautiful as many people may think it is, it is such a literary design, obviously made for text use, that it always seems out of place in display sizes.

29. ^{C}Pointless as you think it will be, this post is actually worth spending the two minutes that it will take to read, because these are so obvious that theyre most often overlooked.

30. ^{C}Crazy as some might think that is, the customer service was incredible and the product even more so.

31. ^{C}Difficult as they say it will be, Inshā’Allāh it will be better than every Ramaḍān before.

32. ^{C}Unsightly as some say it is, wind farming is more sustainable compared to the generating energy using fossil fuel.

33. ^{C}prejudiced as the test seems to say they are.

34. ^{C}Smart though we might think we are, we do not have access to all information, nor are we experts in every field of human endeavor, so we have to rely on others.

35. ^{C}Clean as you think your hands are, they usually have dirt and bacteria from items youve touched.

36. ^{C}Happy as I want you to be and hope you will be, you must yet understand that marriage is Gods design and His purposes must be pursued in order for you to be truly happy.

37. ^{C}Neither as clever nor as interesting as it appears to think it is, The Words maroons its talented stars in an overly complex, dramatically inert literary thriller that’s ultimately a poor substitute for a good book.

38. ^{C}Because as terrible a thought as she knows it is, Irene is just so damn tempting.

39. ^{C}For as small and insignificant as some other luxury and green car brands might want us to think Tesla is, the name keeps popping up all over the news, and even when its about Teslas losses in the courtroom, most of the American public seems to side with Tesla.

40. ^{C}For as independent and fierce as Miley Cyrus wants everyone to think she is, she's actually pretty scared of being alone – and as a result, she's starting to blame herself for Liam Hemsworth's alleged wandering eye.

41. ^{C}So as safe as the CSE promoters want you to believe the process is, there is a significant potential for both loss and inconvenience— enough so that managed municipal, corporate, and government CEFs, REITs, preferred stocks, etc.

42. ^{C}Clever as I thought my nerd humour was, it would appear that this was a bit of a recurring joke among nerd tweeps and just proved once again that there is no original thought.

43. ^{C}As entertaining as we may think we are everyone in the room wants to be in their own area getting ready for kids.

44. ^{C}Ardent, as pretentious and intelligent as you want to keep trying to tell people u are, u should know what AI is and how far Siri is from AI.

45. ^{C}As exceptional as your client wants to think you are, you must come in second to him.

46. ^{C}Because as terrible a thought as she knows it is, Irene is just so damn tempting.

47. ^{C}As odd as I know it sounds I think that adding a few extra bits is part of the experience of this set.

48. ^{C}As smart as I think I am now I know that it is only because I had to walk some long and hard steps to get here.

49. ^{C}It is my understanding (as limited as you seem to think it is) is that trustees (perhaps not the names you've mentioned) were the ones who decided to create an investment office and the parameters of the offices responsibilities.

50. ^{C}As crazy as I thought it sounded I told him it was fine with me.

51. ^{C}As anonymous as I thought I wanted to be, there was something about being there and singled out for anything other than panhandling that felt like a casting call.

52. ^{C}As practical as I can imagine being sighted must be I sometimes think sight can be a very big obstacle.

53. ^{C}However, as secure as many people want to believe blockchain is, it is not without its vulnerabilities.

54. ^{C}As busy as I know Kelly will want to be on her first day at Gonzaga, ... Hoopfest must come first.

55. ^{C}As profound and self-important as Arrival appears to think it is, your humble reviewer found it a yawn-inducing snoozefest that borrowed from every science fiction film from The Day the Earth Stood Still to Close Encounters of the Third Kind to ET to Contact.

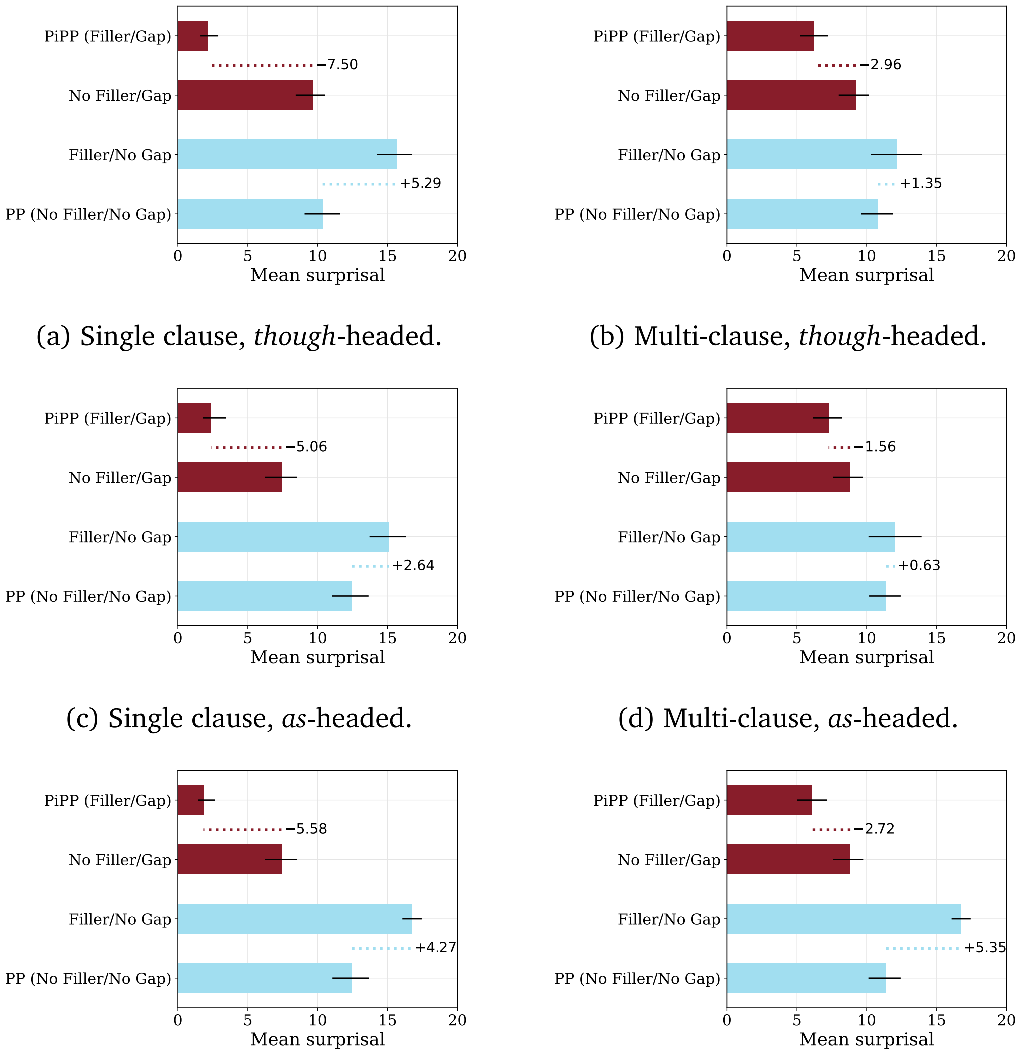
56. ^{C}Well-intentioned though many people may have imagined that the CIA probably thought they were, their foreign-policy operations were confused, duplicitous failures.

57. ^{C}As for planning, as sinister as I think this student thinks our meetings may be, they are really not!

58. ^{C}As much of a downer as I think we both agree the pistols are, for us, do you not find the only thing worse than using one yourself is when someone else in the lobby absolutely dominates with them, when running them akimbo?

F Additional large language model results

Figures 6 and 7 show the results of testing wh-effects in PiPPs using `text-davinci-001` and `text-davinci-003`, respectively.



(a) Single clause, *though*-headed.

(b) Multi-clause, *though*-headed.

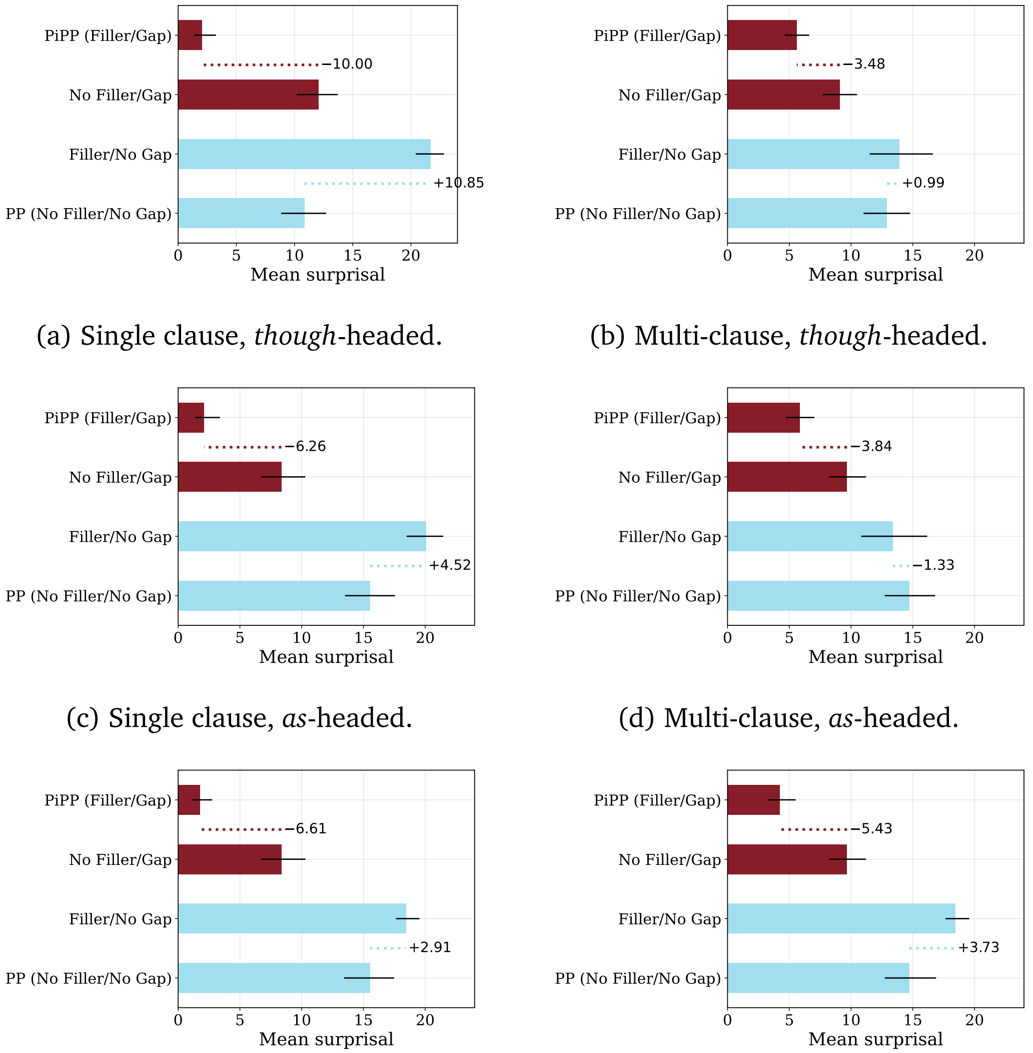
(c) Single clause, *as*-headed.

(d) Multi-clause, *as*-headed.

(e) Single clause, *as*. . . *as*-headed.

(f) Multi-clause, *as*. . . *as*-headed.

Figure 5: Testing wh-effects for GPT-2 large. The models shows the expected effects in all conditions.



(a) Single clause, *though*-headed.

(b) Multi-clause, *though*-headed.

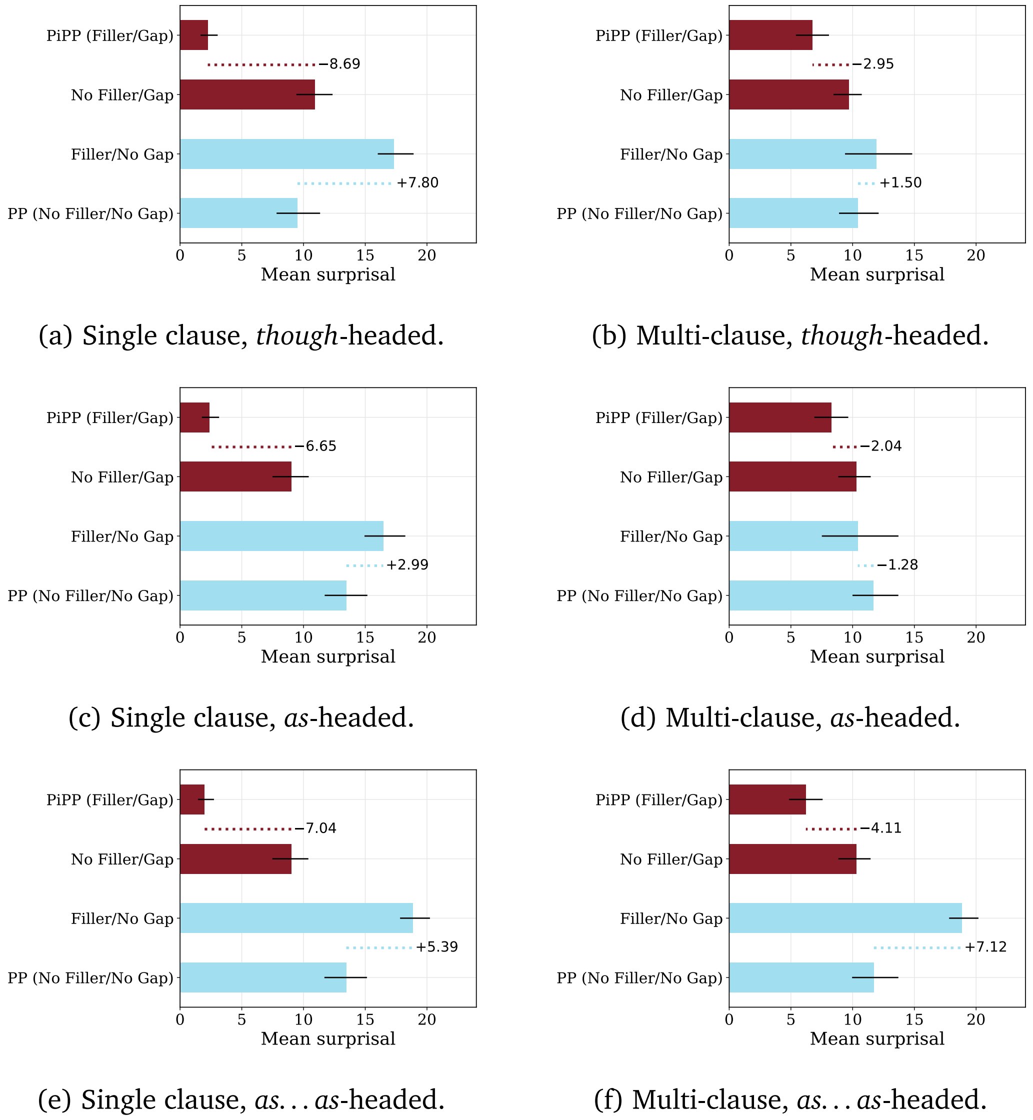
(c) Single clause, *as*-headed.

(d) Multi-clause, *as*-headed.

(e) Single clause, *as*. . . *as*-headed.

(f) Multi-clause, *as*. . . *as*-headed.

Figure 6: Testing wh-effects for the GPT-3 `text-davinci-001` model. The model shows the expected effects everywhere except for the multi-clause *as*-headed condition.



(a) Single clause, *though*-headed.

(b) Multi-clause, *though*-headed.

(c) Single clause, *as*-headed.

(d) Multi-clause, *as*-headed.

(e) Single clause, *as*. . . *as*-headed.

(f) Multi-clause, *as*. . . *as*-headed.

Figure 7: Testing wh-effects for the GPT-3 `text-davinci-003` model. The model shows the expected effects everywhere except for the multi-clause *as*-headed condition.

G LLMs as linguists

Can LLMs reliably transform PPs into PiPPs? This is an unusual task, and so it should not be considered a prerequisite for mastering PiPPs, but a positive result here seems like it would be informative. To begin such an assessment, I ran some small pilots with `text-davinci-001` and `text-davinci-003`. It should be emphasized that, in these experiments, the models are frozen objects. To the extent that they "learn", it is entirely the result of the prompt placing them in a particular temporary state. This is what the NLP literature refers to as *few-shot in-context learning* (Brown et al. 2020).

The primary materials for my pilot are the 33 base sentences from Section 5.3. As before, we can automatically create variants of these basic sentences with different prepositions and different levels of embedding.

The prompt to the LLM includes some high-level instructions, and then it offers some number of demonstrations of the intended behavior: translating PPs into PiPPs. The demonstrations are drawn from the experimental materials, always disjoint from the target item. After the demonstrations, I include the instruction `Now apply the transformation to this input:`, a novel input, and the string `Output:`. Here is a toy example of the prompt, using two invented short examples as demonstrations, and a toy target case:

```
You are an expert grammarian. Your task is to convert the Input
example to a new Output by applying a transformation. Here are
some examples of these transformations:

Input: Though they were happy, they said no.
Output: Happy though they were, they said no.

Input: I am, though it seems odd, friends with a robot.
Output: I am, odd though it seems, friends with a robot.

Now apply the transformation to this input:

Input: Though they felt sad, they smiled.
Output:
```

The model's entire continuation (with peripheral whitespace removed) is taken to be its prediction, and we say the model is correct if and only if its prediction is an exact string match (EM) with the gold PiPP. For the above, the correct output would be *Sad though they felt, they smiled.*

Table 2 summarizes the results. The `text-davinci-001` engine struggles to perform the task, but `text-davinci-003` is outstanding at it. For that model, the assessment is perhaps unfairly strict, as the three cases that were marked as incorrect are the following, in which the model created a well-formed PiPP and happened also to change the position of the entire PiPP in the string:

(43) PP We liked the end of the movie, as they said that we knew that it was tragic.
Gold We liked the end of the movie, as tragic as they said that we knew that it was.
Pred As tragic as they said that we knew that it was, we liked the end of the movie.

(44) PP The proposal is still being assessed, as they said that we knew that it seemed inspired.
Gold The proposal is still being assessed, as inspired as they said that we knew that it seemed.
Pred As inspired as they said that we knew that it seemed, the proposal is still being assessed.

(45) PP They skipped the movie, as they said that we knew that it seemed exciting.
Gold They skipped the movie, as exciting as they said that we knew that it seemed.
Pred As exciting as they said that we knew that it seemed, they skipped the movie.

In these materials, the fronted material is always a single adjective. To assess whether models could perform the PiPP transformation on a wider range of constituents, I created nine additional “stress test” cases. These are included in the code and data release for the paper, as `materials-stress-test.csv`. I repeated the above experiments using these items. With demonstrations drawn from the stress-test examples (always disjoint from the target), `text-davinci-003` gets only 3/9 correct. Essentially the same result obtains (2/9) when the demonstrations are drawn from the basic materials (randomly sampling from different preposition types and different embeddings). This suggests that, in the general case, applying this transformation is challenging for these models – as it would be for many people.

Engine	Preposition	Embedding	Accuracy (EM)
text-davinci-001	as	None	0.70
	as	they said that we knew that	0.64
	though	None	0.48
	though	they said that we knew that	0.55
	as…as	None	0.70
	as…as	they said that we knew that	0.88
text-davinci-003	as	None	0.67
	as	they said that we knew that	0.64
	though	None	0.70
	though	they said that we knew that	0.70
	as…as	None	0.88
	as…as	they said that we knew that	0.91

Table 2: Assessment of model abilities to transform PPs into PiPPs using only few-shot, in-context learning.