Generating event descriptions under syntactic and semantic constraints

Angela Cao[1], Faye Holt[2], Jonas Chan[2], Stephanie Richter[1], Lelia Glass[2], and Aaron Steven White[1]

[1]*University of Rochester*
[2]*Georgia Institute of Technology*

Abstract

With the goal of supporting scalable lexical semantic annotation, analysis, and theorizing, we conduct a comprehensive evaluation of different methods for generating event descriptions under both syntactic constraints—e.g. desired clause structure—and semantic constraints—e.g. desired verb sense. We compare three different methods—(i) manual generation by experts; (ii) sampling from a corpus annotated for syntactic and semantic information; and (iii) sampling from a language model (LM) conditioned on syntactic and semantic information—along three dimensions of the generated event descriptions: (a) naturalness, (b) typicality, and (c) distinctiveness. We find that all methods reliably produce natural, typical, and distinctive event descriptions, but that manual generation continues to produce event descriptions that are more natural, typical, and distinctive than the automated generation methods. We conclude that the automated methods we consider produce event descriptions of sufficient quality for use in downstream annotation and analysis insofar as the methods used for this annotation and analysis are robust to a small amount of degradation in the resulting event descriptions.

1 Introduction

The development and evaluation of an empirically robust lexical semantic generalization requires at least two kinds of data:

(i) a representative sample of lexical items that fall under that generalization;

(ii) for each such lexical item,

(a) a representative sample of the linguistic expressions that each lexical item can (and perhaps, cannot) be used in, along with their context of use;
(b) information about the inferences supported by the lexical item in conjunction with its context.

In this paper, we investigate three ways of satisfying requirement (iia) as a means for supporting downstream lexical semantic annotation, analysis, and theorizing: (1) traditional manual generation of linguistic expressions by experts; (2) sampling linguistic expressions from an annotated corpus; and (3) sampling linguistic expressions from a language model (LM).

Our proximal aim is to assess the quality of the linguistic expressions that are produced by each method when that method is required to enforce specific syntactic and semantic constraints on those lexical items. This proximal aim serves our overarching aim: to assess whether automated generation methods, such as methods (2) and (3), are of sufficiently high quality to use in downstream lexical semantic annotation, analysis, and theorizing in the absence of *post hoc* human correction. Insofar as they are safe to use, these automated methods may provide a means of implementing scalable semantic annotation that allows analysts and theoreticians to better target lexical semantic properties of interest than existing "semantic bleaching" methods (White & Rawlins, 2016, 2018, 2020; An & White, 2020; Moon & White, 2020; Kane *et al.*, 2022).

We focus on sampling sentences in English under heavy syntactic and semantic constraints. This focus on English allows us to satisfy requirement (i) by drawing on existing verb lexicons, such VerbNet (Kipper-Schuler, 2005) and PropBank (Palmer *et al.*, 2005), which classify verbs in terms of their senses and subcategorization properties (among other things). And given that English has substantial resources in terms of both annotated corpora and state-of-the-art language models, this focus additionally allows us to interpret our results as a sort of upper bound on what one can expect in terms of the quality of automatically generated sentences.

In three experiments, we compare each method along three dimensions: (a) how natural the sentences it produces are; (b) how typical the (kinds of) events described by the sentences are; and (c) how distinctive the sentences are. High quality in terms of naturalness is desirable in the sense that unnatural sentences cannot be guaranteed to satisfy particular syntactic and semantic constraints specified by the analyst. For instance, (1) is unnatural because of its ill-formed object.

(1) The runner ran the it.

And while one might be able to recover that the type of event described in (1) is similar to the one described in (2)—e.g. on the basis of the well-formed subject—it is generally more desirable to not require an annotator to perform that sort of recovery, since different annotators may do so in different ways—potentially introducing noise into the measure of interest.

(2) The runner ran the marathon.

High quality in terms of typicality is similarly desirable in that atypical event descriptions cannot be guaranteed to satisfy particular semantic constraints specified by the analyst. For instance, in contrast to (1), the object of (3) is syntactically well-formed; but (3) is clearly atypical of the sorts of running described in (2).

(3) The table ran the marathon.

Depending on the sort of annotation of interest, this atypicality may not matter, but we take it that it is generally more desirable to present annotators with typical examples of a lexical item (cf. Reisinger *et al.*, 2015; White *et al.*, 2016, 2020).

Finally, high quality in terms of distinctivenss is desirable in the sense that the example is well-targeted for the semantic property of interest. For instance, (4) can be reasonably thought of as both a natural and typical description of the same event as (2); however, (4) could just as easily be used to describe the same event as (5).

(4) Someone ran something.

(5) The CEO ran the company.

Thus, (4) is intuitively less distinctive—i.e. more general—than either (5) or (2). As for naturalness and typicality, such generality may or may not be a problem, depending on the property of interest. But there are certainly cases where it could present an issue. For example, if one were interested in assessing whether *run* is telic in a transitive, interpreting (4) as involving the same sense as (3), which is a telic description, gives different results than interpreting (4) as involving the same sense as (5), which is an atelic description.

We discuss the benefits and drawbacks of each generation method in Section 2 before turning, in Section 3, to our specific implementation of each method to generate materials for our three experiments. In Sections 4 to 6, we describe our naturalness, typicality, and distinctiveness experiments, in which we find that the automated methods reliably produce natural, typical, and distinctive sentences, but that manual generation continues to produce sentences that are more natural, typical, and distinctive

	Quality	Effort	Efficiency
Manual	High	High	Low
Corpus	?	Low	Medium
LLM	?	Low	High

Table 1: Preliminary summary of sampling methods for linguistic stimuli.

than these automated generation methods. We conclude, in Section 7, that automated methods are of sufficient quality to use as assistive technologies for scaling research in lexical semantics insofar as downstream annotation and analysis tolerates a small amount of degradation in the linguistic expressions to be annotated and analyzed.

2 Three approaches to generating examples

Each generation method has upsides and downsides (summarised in Table 1). We consider each method—manual generation in Section 2.1, sampling from a corpus in Section 2.2, and sampling from a language model in Section 2.3—alongside its upsides and downsides. For the sake of concreteness, we couch our discussion in terms of generating sentences, since that is our focus in this paper; but our discussion applies equally to generating smaller or larger linguistic expressions.

2.1 Manual generation by experts

On the one hand, manually generated sentences can be carefully controlled for a variety of factors that may influence the inferences supported by such a sentence, such as its clausal and nominal structure, as well as morphology. This fine-grained control over such factors in turn supports apples-to-apples comparisons among lexical items.

For example, in a context where we are interested in understanding the properties of the verb *hit* and related predicates, one may aim to generate sentences like those in (6) as exemplars for further investigation (sentences from Fillmore 1970, p. 127).

(6) a. John hit the tree (with a rock).
b. A rock hit the tree.

These sentences are carefully controlled in the sense that: (i) they are monoclausal and lack temporal or modal adjuncts adjuncts, negative polarity items, etc. that could influence the inferences supported by the sentences or the acceptability; (ii) they contain only common names or simple singular (in)definite noun phrases headed by nouns

that stereotypically enter into events of hitting; and (iii) they contain a verb in a simple tense that does not introduce modality, which could substantially alter the inferences that the sentences support. They can furthermore be interpreted with minimal context, in part because they describe relatively stereotypical situations.

In return for this careful control, it is generally expensive and inefficient to generate sentences—especially in cases where an expert is required to ensure that constraints such as those described above are satisfied. This expense and inefficiency presents a challenge in cases where one is interested in developing broad-coverage lexical semantic generalizations on the basis of a large sample, since it may be infeasible to manually generate a sufficiently large sample for downstream annotation in a reasonable amount of time.

2.2 Sampling from a corpus

A less expensive, more efficient approach to sampling linguistic expressions—common in usage-based approaches—is to draw them from a corpus (see Katz, 2019, for a review). For example, to study *hit* using this method, we might sample sentences like those in (7) from a corpus like the Pushshift Reddit dataset (Baumgartner *et al.*, 2020).

(7) a. This is something that really hit me hard when I started working at a school.
b. When coal regulations hit here, there were bread lines.

The main challenge in using such data is that the resulting sentences are not controlled for factors that may influence the inferences supported by an expression. As exemplified in (7), corpus data are often multi-clausal and grammatically complex—perhaps in ways orthogonal to one's purpose. They furthermore often require a rich context for their interpretation, and it is not always clear how much extrasentential context one must retain to ensure that the sentence is interpretable—or if such context is not provided, how the reader infers a likely context on the basis of the sentence itself and what effects that has on measures of interest. For example, in (7a), one has to infer antecedents for several pronouns and interpret *hit* as abstract rather than physical, which may not be the sense of interest.

Additional control can be imposed on sentences sampled from a corpus by using syntactic and semantic annotations to filter the sample. For instance, if one is interested in sampling only examples of *hit* in a transitive clause under its sense of physical contact, syntactic annotations might be combined with word sense annotations to find only sentences that satisfy the relevant constraints.

But this approach gives rise to a further challenge: as additional constraints are imposed on a sample, sentences satisfying those constraints become fewer and further between, and thus the size of the corpus must grow to ensure a sample of consistent size. This growth may need to be potentially quite substantial given that lexical sentences are power law distributed (Zipf 1936, 1949, see Piantadosi 2014 for a review), meaning that truly massive corpora may be required for ensuring that sufficient sample sizes for lower frequency lexical sentences are achieved.[1] Thus, while sampling from a corpus is certainly less expensive than manual generation, its efficiency is dependent on the desired level of control over the syntax and semantics of the sampled sentences.

One way to deal with the challenge posed by sparsity is to combine corpus sampling, constrained by syntactic and semantic information, with post-processing. For instance, (8a) is an example from the Pushshift Reddit dataset that we extracted by looking for instances of *hit* in a transitive construction—a relatively light constraint retaining many matches. In a case where we want an example of *hit* satisfying the heavier constraints we can impose on manually generated sentences, we can then use an automatic procedure that makes use of the sentence's syntactic parse to yield an item such as (8b).

(8) a. **Raw:** I feel like all the New Years resolutioners must hit the gym on 2 January as it was this morning.
b. **Post-edited:** The resolutioners hit the gym.

Insofar as the annotations used to enforce the constraints are of high quality, this procedure will definitionally produce sentences that satisfy the relevant constraints, but it is not known whether these sentences are valid in the same way one can ensure that manually generated sentences are. A main aim of this paper is to assess the quality of the sentences that result from a method of this form.

2.3 Sampling from a language model

An alternative way to deal with the challenge posed by sparsity is to work with a compressed form of a corpus, such as a language model (LM). An LM describes the distribution of strings in a corpus as a probability distribution p_θ with parameters θ. These parameters are estimated (roughly) by obtaining those for which p_θ assigns

[1] One way to deal with this issue is to simply ignore low-frequency lexical items. But one does this at their own peril. See White & Rawlins 2018 for evidence that focusing too heavily on high-frequency items has led to incorrect lexical semantic generalizations.

maximum likelihood to the strings in the corpus.[2]

The upshot of describing the distribution of strings in a corpus as a probability distribution is that, insofar as one knows how to sample from $p_{\boldsymbol{\theta}}$, the LM may be used to (noisily) sample from the corpus that it was estimated on. Assuming that $p_{\boldsymbol{\theta}}$ furthermore assigns non-zero probability to strings that satisfy the desired syntactic and semantic constraints, one can sample sentences that satisfy those constraints from:

$$\bar{p}_{\langle \boldsymbol{\psi}, \boldsymbol{\theta} \rangle}(w_1 w_2 \ldots w_n \mid \textsf{constraints}) \propto p'_{\boldsymbol{\psi}}(\textsf{constraints} \mid w_1 w_2 \ldots w_n) \times p_{\boldsymbol{\theta}}(w_1 w_2 \ldots w_n)$$

One generic way to implement sampling from $\bar{p}_{\langle \boldsymbol{\psi}, \boldsymbol{\theta} \rangle}$ is rejection sampling: (i) sample a string $w_1 w_2 \ldots w_n$ from $p_{\boldsymbol{\theta}}$; and (ii) accept that sample with probability $p'_{\boldsymbol{\psi}}(\textsf{constraints} \mid w_1 w_2 \ldots w_n)$. A major issue with this approach is that is can be extremely inefficient—potentially far more inefficient than simply sampling from the corpus itself—since the vast majority of samples drawn from $p_{\boldsymbol{\theta}}$ will not satisfy the constraints.

At least two approaches can be used to mitigate this inefficiency: (a) sampling from a language model conditioned on a "prompt" that encodes the constraints (Radford *et al.*, 2018; Brown *et al.*, 2020, i.a.); and (b) modifying the language model's probabilities so that strings that satisfy the constraints have higher probability (*constrained sampling*; Holtzman *et al.*, 2018; Dathathri *et al.*, 2020; Yang & Klein, 2021, *i.a.*).[3] Both approaches take advantage of the fact that $p_{\boldsymbol{\theta}}$ is decomposable as:

$$p_{\boldsymbol{\theta}}(w_1 w_2 \ldots w_n) \equiv q_{\boldsymbol{\theta}}(w_1) \times q_{\boldsymbol{\theta}}(w_2 \mid w_1) \times \ldots \times q_{\boldsymbol{\theta}}(w_N \mid w_1 \ldots w_{N-1})$$

where $q_{\boldsymbol{\theta}}$ is a probability distribution on lexical items conditioned on strings. A string can then be sampled by incrementally sampling w_i from $q_{\boldsymbol{\theta}}$ conditioned on $w_1 \ldots w_{i-1}$.

2.3.1 Sampling from a language model conditioned on a prompt

The LM prompting approach encodes the constraints to be satisfied as natural language strings. For instance, to enforce the semantic constraint that a sentence sampled from an LM contain the verb *hit* in the sense of *strike*, one might sample from $q_{\boldsymbol{\theta}}$ conditioned on (9), where (9) is simply viewed as a string that the LM itself could have

[2]Many contemporary language models are estimated using methods that go beyond maximum likelihood estimation. A simple case of this is the incorporation of regularization terms. Another, very common, more complex case involves *preference tuning*, e.g., using reinforcement learning from human feedback (Christiano *et al.* 2017; Stiennon *et al.* 2020; Ouyang *et al.* 2022, i.a.).

[3]The literatures on both prompting and constrained sampling are vast, and there are many ways to implement both. We focus here on relatively simple variants that do not require us to change anything about the underlying language model—or indeed, train a new system at all—since we believe this setting is the most realistic for most linguists.

generated.

(9) The following is an example of a sentence that contains the verb "hit" in its sense meaning "strike":

The sentences in (10) are sampled from the LM `llama-2-13B` (Touvron *et al.*, 2023) conditioned on this prompt.

(10) a. The baseball hit the bat.
b. He hit the ball out of the park.
c. Joe hit me with his car.
d. John hit me on the head.

Anecdotally, this approach works surprisingly well for simple semantic constraints, such as the one expressed in (9). However, it can be difficult to know exactly how to specify syntactic and morphological constraints in a general way. For example, perhaps we want all of the arguments to be definite noun phrases (NPs)—e.g. because we want the sentences to be maximally evocative of the relevant sense and also make more sense out of context. Or, perhaps we do not want the prepositional phrases (PPs) that appear in (10b)–(10d)—e.g. because the structure in (10b)–(10d) could produce importantly different inferences than the simple transitive.

2.3.2 Constrained sampling

Enforcing such constraints is where the constrained sampling approach shines. In constrained sampling, one specifies constraints on the strings to be sampled as an auxiliary probability distribution on those strings.[4] These constraints can in principle be arbitrarily complex—e.g. requiring access to an entire string (Holtzman *et al.*, 2018, i.a.). But as discussed above, this complexity can make sampling difficult—or at the very least, require the estimation of additional models, which we aim to avoid.

To mitigate this difficulty, we consider only constraints that are themselves decomposable into a conditional distribution c_γ with parameters γ. These constraints can then be combined with the incremental sampling procedure by sampling from a

[4]Constrained sampling is closely related to *constrained decoding*, which is a widely-used approach in structured prediction that attempts to obtain an output with maximal probability (or more generally, score), rather than sampling outputs from a constrained distribution. See Smith 2011 for a general overview of structured prediction for linguistic data and Deutsch *et al.* 2019; Shin *et al.* 2021 and references therein on constrained decoding. A main reason to use constrained sampling, rather than constrained decoding, is that the resulting strings tend to be higher quality (Holtzman *et al.*, 2019).

constrained distribution $\bar{q}_{\langle\boldsymbol{\theta},\gamma\rangle}$, rather than $q_{\boldsymbol{\theta}}$ itself.

$$\bar{q}_{\langle\boldsymbol{\theta},\boldsymbol{\gamma}\rangle}(w_i \mid w_1 \dots w_{i-1}) \propto q_{\boldsymbol{\theta}}(w_i \mid w_1 \dots w_{i-1}) \times c_{\boldsymbol{\gamma}}(w_i \mid w_1 \dots w_{i-1})$$

The distribution c_γ can take a variety of forms. In the case where one wants to enforce syntactic constraints, a natural way to state them is in terms of a probabilistic context free grammar (PCFG), where $\boldsymbol{\gamma}$ gives rule probabilities. Then, $c_{\boldsymbol{\gamma}}(w_i \mid w_1 \dots w_{i-1})$ can be computed using, e.g., an Earley parser (Earley, 1970; Stolcke, 1995).[5]

For instance, if one wants to enforce the syntactic constraints discussed in Section 2.1—that all arguments be definite NPs and that the sorts of prepositional phrases (PPs) that appear in (10b)–(10d) be excluded—one might define c_γ in terms of (11).

(11)
```
S -> NP VP
NP -> D N
VP -> V NP
V -> hit
D -> the
N -> .+
```

In this grammar, the `S`, `NP`, `VP`, `V`, and `D` rules would all necessarily have probability 1—and thus they impose hard constraints on the samples—and `N -> .+` is to be interpreted such that `N` can be rewritten equiprobably as any non-empty string. The latter would be too permissive if this grammar were intended to model transitive clauses headed by *hit*, but because the language model provides information about words that are likely to come after *the*, further constraint is unnecessary.

The result of defining c_γ in terms of (11) is to enforce that (12).

(12)
a. $\bar{q}_{\langle\boldsymbol{\theta},\gamma\rangle}(\text{the}) = 1$
   because $c_{\boldsymbol{\gamma}}(\text{the}) = 1$
b. $\bar{q}_{\langle\boldsymbol{\theta},\gamma\rangle}(w \mid \text{the}) = q_{\boldsymbol{\theta}}(w \mid \text{the})$
   because $c_{\boldsymbol{\gamma}}(w \mid \text{the}) \propto 1$ for all w
c. $\bar{q}_{\langle\boldsymbol{\theta},\gamma\rangle}(\text{hit} \mid \text{the}, w) = 1$
   because $c_{\boldsymbol{\gamma}}(\text{hit} \mid \text{the}, w) = 1$ for all w
d. $\bar{q}_{\langle\boldsymbol{\theta},\gamma\rangle}(\text{the} \mid \text{the}, w, \text{hit}) = 1$
   because $c_{\boldsymbol{\gamma}}(\text{the} \mid \text{the}, w, \text{hit}) = 1$ for all w

[5]This use of probabilistic Earley parsers is well-known in the psycholinguistics literature for its use in evaluating information-theoretic models of sentence processing (see Hale, 2001; Levy, 2008, *et seq*). No stock should be placed in our suggestion of a PCFG over a more expressive formalism, such as a probabilistic linear context free rewriting system (Kato *et al.*, 2006; Kallmeyer & Maier, 2013). The main criterion that must be satisfied is that a conditional distribution over the next word be computable.

e. $\bar{q}_{\langle \boldsymbol{\theta}, \boldsymbol{\gamma} \rangle}(w' \mid \text{the}, w, \text{hit}, \text{the}) = q_{\boldsymbol{\theta}}(w' \mid \text{the}, w, \text{hit}, \text{the})$
because $c_{\boldsymbol{\gamma}}(w' \mid \text{the}, w, \text{hit}, \text{the}) \propto 1$ for all w, w'

The main downside of this approach to constrained sampling is that the distribution over earlier words—e.g. the noun in the subject of *hit*—cannot be constrained by later words—e.g. *hit* itself and the noun coming after *hit*.[6] We mitigate this downside in two ways: (i) by combining prompting and constrained sampling as a means to provide information about the verb that will be generated before that verb's subject is generated; and (ii) by generating multiple samples that are then reranked by their probability under the language model (discussed in Section 3).

2.3.3 Combining prompting and constrained sampling

Prompting and constrained sampling can be combined quite easily. However, it remains an open question whether—under the sorts of heavy constraints lexical semanticists work—the sentences sampled using this method are valid in the same way one can ensure that manually generated sentences are. For instance, does prompting a language model with a gloss of a verb's sense actually produce good examples of that verb in that sense? (14) shows four sentences sampled from `llama-2-13b` conditioned on the prompt in (13). While (14a) is a good example of the *reach, encounter* sense of *hit*, (14b)–(14d) are better examples of the *strike* sense.[7]

(13) The following is an example of a sentence that contains the verb "hit" in its sense meaning "reach, encounter":

(14) a. We hit the road at dawn.
b. The ball hit him squarely on his forehead.
c. He hit his head on the door frame.
d. She hit her head on the wall.

[6] An alternative route to constrained sampling that does not have this downside is to use a masked language model, which is trained to predict the identity of elements $w_{i_1}, \ldots, w_{i_k}$ in a string $w_1 w_2 \ldots w_n$ given all elements of the string besides $w_{i_1}, \ldots, w_{i_k}$ (Devlin *et al.*, 2019). For instance, one could in principle sample w_2 from $p(\cdot \mid \text{the}, -, \text{hit}, \text{the}, -)$ and then w_5 from $p(\cdot \mid \text{the}, w_2, \text{hit}, \text{the}, -)$. We do not take this route for two reasons: (i) anecdotally, using a masked language model—specifically, RoBERTa (Liu *et al.*, 2019)—produced worse samples than an autoregressive language model like `llama-2-13b`; and likely relatedly, (ii) the literature has largely abandoned masked language modeling in recent years, meaning that the largest—and therefore, generally most performant—language models are autoregressive. Thus, we deemed it more practical to focus on autoregressive models.

[7] Both of these sense glosses come from PropBank's manually constructed frame file for *hit*: `hit.01` (*strike*) and `hit.02` (*reach, encounter*).

3 Implementing the three approaches

To compare the three approaches discussed in Section 2 along our three dimensions of interest, we use each method to generate sentences constructed from a well-controlled set of verbs. All sentences are generated under the constraints specified in (15).

(15) a. The sentence must be a monoclausal transitive of the form `NP V NP`.
b. The verb must be in its simple past tense form.
c. The subject and object of the transitive must be definite noun phrases of the form `the N`.
d. Given a verb and a sense of that verb, the interpretation of the sentence must be compatible with that sense. Assuming that the sense is itself compatible with a transitive, this compatibility is solely determined by the nouns in the subject and object.

We describe how we select verbs for inclusion in our sentences in Section 3.1 and ensure that all constraints can be satisfied for those verbs. In Sections 3.2 to 3.4, we describe how we implement the generation approaches laid out in Section 2.

3.1 Choosing verbs

Verbs were chosen with reference to their senses in the PropBank lexicon (Palmer *et al.*, 2005) and their classification according to Levin 1993, as encoded in the VerbNet lexicon (Kipper-Schuler, 2005), which groups verbs by their syntactic behavior—e.g. VerbNet's `hit-18.1` class, which includes *beat*, *kick*, *smash*, and so on.[8] Using these resources, we selected 32 triplets of verbs (96 verbs) reflecting the criteria in (16).[9] Table 2 gives examples of three such triplets.

(16) a. All verbs in the triplet share a Levin class in their transitive form.
b. One verb in the triplet is polysemous in its transitive form according to PropBank, with 3-5 PropBank senses in its transitive form.
c. One verb is monosemous in its transitive form according to PropBank, with 1 PropBank sense in its transitive form.

[8] VerbNet was originally conceived as a digitization of Levin's classification. VerbNet has expanded beyond this original classification (Kipper *et al.*, 2006), but for the subset of classes that correspond to ones in the original classification, the numerical identifier associated with the class corresponds to the (sub)section of Levin 1993 in which that class is discussed.

[9] We take the number of senses that PropBank associates with a particular verb as a rough proxy for how polysemous it is. See Hovy *et al.* 2006 on data-driven approaches to validating PropBank senses using interannotator agreement.

VerbNet Class	Polysemous	Monosemous	Calibration
hit-18.1	hit	kick	smash
judgment-33	abuse	insult	mock
clear-10.3	clear	clean	drain

Table 2: Example verb triplets used in our study.

d. One verb, which we use for our manually generated and calibration sentences, has 1 or 2 senses in its transitive form according PropBank.
e. All verbs in the triplet were manually judged as very similar in meaning, on some sense of the polysemous members.

We choose both polysemous (e.g. *hit*) and monosemous (e.g. *kick*) verbs from each class in order to explore how well each automated method is able to generate sense-distinct sentences in cases where a verb has multiple senses. We select a third *calibration* verb (e.g. *smash*) in order to produce manually generated sentences that we use for two purposes: (i) to assess the extent to which experts can reliably generate sentences that vary with respect to naturalness, typicality, and distinctiveness; and (ii) to calibrate participants judgments in each of the three tasks described in Sections 4 to 6. We describe both uses in more detail in Sections 4 to 6.

3.2 Manual generation by experts

We wrote sentences by hand for our 32 calibration verbs. For our first set of calibration sentences, we wrote four sentences for each verb: one that we deemed to be a *natural* description of a *typical* situation; one that we deemed to be a *natural* description of an *atypical* situation; one that we deemed to be an *unnatural* description of a *typical* situation; and one that we deemed to be an *unnatural* description of an *atypical* situation. This procedure resulted in 124 manually generated sentences within our first set.

In more detail, for a calibration verb, we first chose a subject and object deemed *natural* and *typical* when used with the verb in question. The distinction we draw between these two dimensions is that natural sentences follow all structural rules of English, while typical sentences use content nouns that play an expected role in the situation described by the verb—e.g. cooks commonly *smash* potatoes as in (17).

(17) The cook smashed the potatoes.

Next, we changed either the subject or object of the natural and typical sentence to an-

other noun phrase such that the result is *natural* but *atypical*. The atypical sentences use content nouns that play a surprising role in that situation—e.g. it is unusual for a strawberry to *smash* a potato, as in (18).

(18) The strawberry smashed the potatoes.

To generate the *unnatural* but *typical* sentence, we took the natural and typical sentence and change the noun phrase that was unchanged for the natural but atypical sentence. Unnatural sentences contain syntactic violations—e.g. combining the pronoun *them* with a definite determiner, as in (19).

(19) The cook smashed the them.

Finally, the *unnatural* and *atypical* sentence takes the two new noun phrases introduced by (18) and (19) and replaces those in (17)—exemplified in (20).

(20) The strawberry smashed the them.

There are an equal number of sentences in which the subject is replaced for the natural and atypical sentences as there are sentences in which the object is replaced for the natural and atypical sentences.

We additionally generated a second set of calibration sentences for use in our experiment investigating distinctiveness (Section 6). This experiment asks participants to judge pairs of sentences on the basis of the similarity of the events they describe. A crucial feature that we aim for our manually generated pairs to have is that, in half the pairs, the verb contained in both sentences has the same sense, and in the other half, it has a clearly distinct sense.

We started with the natural and typical sentences written for our first set of calibration sentences. We then added to this set, for each natural and typical sentences, another sentence that used the *same* sense as the natural and typical sentence, such as in (21).

(21) a. The cook smashed the potatoes.
b. The shoe smashed the bug.

We then generated another sentence that used what we deemed to be a different sense compared to that of the reference sentence, as in (22).

(22) a. The cook smashed the potatoes.
b. The startup smashed the competition.

This process results in 64 pairs of sentences, in which 32 are same-sense pairs and 32 are different-sense pairs.

3.3 Sampling from a corpus

To implement sampling from a corpus, we sampled comments from Reddit found in the PushShift dataset (Baumgartner *et al.*, 2020). First, we automatically sense-tagged these raw sentences using a highly performant sense-tagger (Shi & Lin, 2019; Orlando, 2020) trained on the PropBank annotations in the version of OntoNotes (Hovy *et al.*, 2006) used in the CoNLL-2012 shared task (Pradhan *et al.*, 2012). For example, (8a) from Section 2 was sense-tagged to represent the *turn to, go to* sense of *hit* in PropBank, while (23a) was sense-tagged as *reach, encounter*.

(23) a. **Raw:** Like the sensation was so strong, I went into what felt like shock, a tingle up my spine, and my body hit the floor.
b. **Post-edited:** The body hit the floor.

In total, we tagged approximately ∼129 million comments (∼3.5 billion words).

As noted in Section 2, it is very rare to find sentences that perfectly satisfy the constraints in (15): many sentences are multi-clausal, use complex grammatical tense and/or aspect, and contain many pronouns. To deal with this issue, we use the SpaCy (v3.5.3; Honnibal & Montani, 2017) dependency parser (`en_core_web_lg`) to filter data to sentences containing our target verbs whose dependents include both a subject and an object noun with determiners (excluding highly bleached noun phrases such as *a lot*). We then automatically edit these sentences—changing all verb tenses to the past, replacing all determiners with *the* and removing extraneous clauses to fit the desired form. (8b) and (23b) show the result of this procedure for (8a) and (23a), respectively.

The result of this editing procedure is not guaranteed to be a particularly natural sentence. As a heuristic for finding the most natural sentences, we computed each edited sentence's surprisal using GPT-2 (Radford *et al.*, 2018) and manually reviewed the 10 lowest surprisal sentences exemplifying each sense, for all target verbs (1,032 sentences total). Of these, we disqualified sentences which were misparsed—e.g. *B.C.* (British Columbia) in (24a)—incorrectly sense-tagged—e.g. `assert.02` assigned to *belt* in (24b)—or which used jargon/esoteric NPs—e.g. the proper name *Kulaks* in (24c).

(24) a. The B.C. burned the land.
b. The fanbase belted the anthem.
c. The Kulaks burned the crops.

PropBank sense gloss	Example from Reddit
`hit.01` (strike)	The bullets hit the wall.
`hit.02` (reach, encounter)	The tape hit the news.
`hit.03` (go to, turn to)	The resolutioners hit the gym.

Table 3: Examples for *hit* found in sense-tagged data from Reddit.

The fact that we use a manual selection procedure means that the sentences are not selected purely automatically. Our aim in performing manual selection was to simulate the result of an automatic procedure whose inputs were perfectly annotated. As such, our results in Sections 4 to 6 must be interpreted as an upper bound on the quality of examples generated from a corpus.

After this filtering step, we selected the four sentences for each sense with the lowest surprisal (based on the post-edited version of the sentences). If there were less than 4 sentences fitting these criteria, we took the top 2–3.[10] Table 3 shows some examples of sentences representing three senses of *hit*. Our final set of corpus-generated sentences comprises 367 sentences.

3.4 Sampling from a language model

To implement sampling from an LM, we sampled sentences from a quantized variant of `llama-2-13b`[11] using nucleus sampling (Holtzman *et al.*, 2019).[12] Sampling was also conditioned on the prompt template in (25), where {{`VERB`}} is replaced by the root form

[10]Some senses of verbs are only found in particular dialects and/or genres of English. Reddit features primarily US English, and as a consequence, some rare senses of verbs are not represented at all within our data. For example, the verb *pinch* has a common sense `pinch.01: squeeze tightly to cause pain` which is widely used by all speakers of English. A more uncommon sense is `pinch.02: slangy steal`, which is much more likely to appear in less common varieties of English, and in different genres. In the case where the LM-generated sentences represent such senses (described in Section 3.4) that Reddit did not, we simply do not use a Reddit parallel for those senses in our experiments.

[11]A quantized variant of an LM is one that represents the parameters of the original LM using a lower precision numeric representation than the LM was originally trained with. Quantization makes LMs substantially more efficient to sample from. Specifically, we used a model (`Q5_K_M` from `https://huggingface.co/TheBloke/Llama-2-13B-GGUF`) with 5-bit quantization (type 1) and super-blocks containing 8 blocks, each block having 32 weights and scales and mins quantized with 6 bits.

[12]We used constrained sampling to produce at most 32 tokens and used a top-p of 0.95, a min-p of 0.05, a typical-p of 1.0, a repeat penalty of 1.1, a tail-free sampling parameter of 1.0, a target cross-entropy of 5.0, a learning rate used to update mu of 0.1, a top-k of 40, and starting temperature of 0.8 for each verb-sense. Most of these settings are the default from `https://llama-cpp-python.readthedocs.io/en/latest/api-reference/`. To generate unique sentences, we incremented the seed by 1 on successive calls to the sampler. If after 100 increments in seed, a sufficient number of unique sentences had not been sampled—as noted below, we aimed for 10 for every verb-sense pair—the temperature was increased by 0.1 and sampling continued. This procedure was iterated until either a sufficient number of unique sentences were sampled or 100 different seeds at 100 different temperatures were attempted.

PropBank sense	Example from LM
`hit.01` (strike)	The baseball hit the fence.
`hit.02` (reach, encounter)	The ball hit the wall.
`hit.03` (go to, turn to)	The road hit the lakefront.

Table 4: Examples for *hit* given by the LM when prompted with PropBank's sense glosses.

of the verb of interest and {{SENSE_GLOSS}} is replaced by a gloss of one of that verb's senses, and was constrained by the grammar in (11), replacing the rule `V -> hit` with the simple past tense form of the verb of interest (see Section 2.3).

(25) An example of a sentence containing the verb "{{VERB}}" in the sense "{{SENSE_GLOSS}}":

As in the manual and corpus-based generation methods, we aim to generate four sentences per verb sense. To achieve this, we sample 10 sentences per verb-sense then rank those sentences by surprisal.[13] Then, we select the four with the lowest surprisal. This approach was necessary in order to eliminate nonsense generations, such as (26).[14]

(26) The waiter passed the saltedbuttertohiscustomerintheplasticdishandthencollecteditfromhimafterhewasdonewithit.

We derive the sense glosses in one of two ways: (i) we use the sense glosses provided in PropBank directly; and (ii) we produce sense glosses from an LM. The idea behind comparing these two alternatives is to understand how much one can rely solely on an LM to generate examples without the need for external resources like PropBank. Table 4 shows example generations for the verb *hit*, conditioned on the senses of *hit* found in PropBank.

To implement the second method of deriving sense glosses, we sampled sense glosses from a quantized version of `llama-2-13b-chat`.[15] This method consisted of two stages:

[13] We were unable to sample 10 unique sentences for the one sense of *see*—`see.09` (*visit/consultation by medical professional*) from Propbank—for which we could only obtain four unique sentences despite attempts using 100 different seeds at 100+ different temperatures.

[14] Nonsensical run-on sentences such as this arise because of the LM's strong proclivity towards a token that does not match our grammar, which the LM then forces to fit into our grammar by appending more tokens until it does.

[15] We used the `gptq-8bit-128g-actorder_True` from https://huggingface.co/TheBloke/Llama-2-13B-chat-GPTQ. We set the maximum tokens was set to 512, the minimum length to 10, the top-k to 50, and try temperatures of 0.7, 0.8, and 0.9. Most of these parameters are the defaults from https://llama-cpp-python.readthedocs.io/en/latest/api-reference/.

(i) prompting the LM to classify verbs into monosemous and polysemous verbs; (ii) prompting the LM (a) to produce a single sense gloss for those verbs it classified as monsemous; or (b) to produce an enumeration of sense glosses for those verbs it classified as polysemous.

We took this two-stage approach because we found that the LM tended to offer multiple senses for all verbs—even those listed as monosemous in PropBank. One might posit that this pattern is evidence that PropBank "undercounts" senses—and that may well be the case (cf. Petersen & Potts, 2023). But even if so, it seems likely that the number of senses posited by the LM for a verb would be correlated with the number found in PropBank, and we found that it was not across the vast majority of draws from the LM.

This behavior is potentially problematic for comparing manually generated sense lexicons, like PropBank's, and sense lexicons generated from an LM because the quality of the LM-generated lexicon is likely highly sensitive to the form of the prompt and other random factors. We cannot completely mitigate these factors, but we attempt to control for at least some of them—specifically, the relative number of senses associated with a verb—by synthesizing the results across multiple prompts in a way that aligns best with PropBank.[16] Our aim here is to focus our analysis mainly on the quality of the glosses produced by the LM, rather than the number.

3.4.1 Stage 1: Monosemy v. polysemy

To implement the first stage and hopefully rein in the LLM's tendency for sense proliferation, we derived a measure of monosemy/polysemy for each verb from the eight prompt template variants found in (27). We then combined these measures to best predict monosemy/polysemy in PropBank.

(27)
a. Does the verb "{{VERB}}" have only one sense when used in a transitive clause?
b. Does the verb "{{VERB}}" have only one possible meaning when used in a transitive clause?
c. Does the verb "{{VERB}}" have only one sense when used in a transitive clause? Only answer with YES or NO.
d. Does the verb "{{VERB}}" have only one possible meaning when used in a transitive clause? Only answer with YES or NO.
e. Does the verb "{{VERB}}" have MORE THAN one distinct meaning when used in a transitive clause?

[16] This procedure is relatively generic and could be used to compare with alternative sense lexicons.

f. Does the verb "{{VERB}}" have more than one distinct meaning when used in a transitive clause?
g. When used in a transitive clause, does the verb "{{VERB}}" have ONE meaning, or MORE THAN ONE distinct meaning?
h. When used in a transitive clause, does the verb "{{VERB}}" have one meaning, or more than one distinct meaning?

We derived this measure by computing the log-odds $\log p(\textsc{yes}) - \log p(\textsc{no})$, where $p(\textsc{yes})$ is the probability that the first output token is *yes* (or some capitalized variant)—and similarly for *no*. We then trained a support vector machine (SVM) to predict whether a verb is monosemous in PropBank using these eight measures as predictors, which we validated using nested cross-validation.[17]

This classifier achieves an accuracy of 0.55 in predicting whether a verb that was held-out in the cross-validation is monosemous or polysemous according to Propbank.[18] We refit the classifier with the optimized hyperparameters to the entire dataset, then use this classifier to predict monosemy v. polysemy for use in the second stage. The accuracy of this refit classifier is 0.77.[19] In total, 30 verbs were labeled as monosemous (including *allow*, *box*, and *button*) and 34 verbs were labelled as polysemous (including *abuse*, *beat*, and *break*).

3.4.2 Stage 2: Sense gloss generation

We use the output of the classifier developed in the first stage to determine whether to generate one sense gloss for a verb or multiple. In cases, where we aim to generate just one gloss, we use the prompt in (28).

(28) Please describe the one possible sense of the verb "{{VERB}}" when it is used in a transitive clause.

[17]This approach splits the data into outer test folds and train-dev folds. Iterating through each pairing of train-dev and test folds, we further split train-dev folds into inner development folds and train folds. We train various model on train folds and evaluate on development folds to receive a score—performing grid search over kernel type (`linear`, `rbf` with `sklearn`'s default kernel coefficient) and regularization parameter (0.01, 0.1, 1.0, 2.0, 5.0, 10.0)—and to select the best model. The final score indicates performance on the outer folds of the model selected on the inner folds.

[18]We provide this accuracy mainly as a point of information. Our goal in building this classifier is not high accuracy, but rather optimal alignment between the number of senses posited in a manually constructed lexicon and the number posited by the LM-based method. That optimal alignment may not be particularly good, as in this case.

[19]This accuracy is not informative of the classifiers performance, since it is evaluated on the data it was trained on. Again, our goal here is optimal alignment with PropBank, not evaluation of the model on monosemy v. polysemy prediction.

LM's own sense gloss	Example from LLM
strike or collide with something	The ball hit the wall.
reach or affect something	The storm hit the coastline.
play a musical note	The piano hit the notes.
be popular or successful	The song hit the charts.
use violence or force	The police hit the rioters.
cause a reaction or response	The movie hit the critics.
fulfill a requirement or expectation	The candidate hit the mark.

Table 5: Examples for *hit* given by the LLM with the LLM's own sense glosses.

In cases where we need to generate multiple glosses, we again attempt to align the number of glosses with the number of glosses in PropBank. we start with the base prompt in (29). We then append to this prompt each possible combination of 1–4 prompts—always in the order given here.

(29) Describe and enumerate the distinct possible senses of the verb "{{VERB}}" when it is found in a transitive clause.

(30)
a. For example, if you were given the verb "administrate", you should respond with "manage" because "administrate" has one transitive sense.
b. For example, if you were given the verb "abandon", you should respond with "1. leave behind; 2. exchange; 3. surrender, give over" since "abandon" has three transitive senses.
c. For example, the verb "jump" has five senses because it has multiple possible meanings when it is used, so you should output something like "1. stock prices, increase, 2. be excited for an opportunity, getting there first, 3. physically or metaphorically leap, physical motion, 4. to escape, bail out, 5. attack, gangsta style".
d. Ensure that the sense description(s) can stand alone and do not depend on being the synonym of some other verb.

The examples of sense descriptions are taken verbatim from a verb in Propbank for which we are not generating senses in order to better align the sense lexicons.

We parse the output of these prompts to retrieve an integer value for each prompt-verb pair. For each prompt, we calculate the mean-absolute error between its sense counts compared to Propbank's to select the prompt whose sense count distribution best matches Propbank's. We use bootstrapping to estimate confidence intervals for the mean absolute error differences between the best prompt and other prompts. This gives us a set of best prompts, of which we choose the one with the shortest length and

Propbank's sense gloss	Example from LLM
hit.01: strike	The pitcher hit the ball.
hit.02: reach, encounter	The ball hit the wall.
hit.03: go to, turn to	The road hit the town.

Table 6: Examples for *hit* given by the LLM with Propbank's transitive sense glosses.

lowest temperature. This yields the final prompt in (31).

(31) Describe and enumerate the distinct possible senses of the verb "{{VERB}}" when it is found in a transitive clause. Feel free to give only one sense if it only has one possible meaning. For example, if you were given the verb "administrate", you should respond with "manage" because "administrate" has one transitive sense.

We suspect that (31) is successful because its only example is a verb said to be monosemous, combating the LM's tendency to offer many sense glosses. Using this prompt, our LM gives a median of one sense to the verbs that PropBank categorizes as monosemous, and a median of five senses to those that PropBank categorizes as polysemous (compared to PropBank's median of three senses for such verbs). Table 5 shows example generations for the verb *hit*, conditioned on the LM-generated senses of *hit*.

4 Experiment 1: Naturalness

Our naturalness experiment was separated into two subexperiments: one focused on the manually generated sentences (Section 4.2) and the other focused on the automatically generated sentences (Section 4.3).

4.1 Instructions and practice sentences

In both subexperiments, participants are asked to rate the naturalness of each sentence on a continuous scale ranging from *extremely unnatural* to *perfectly natural*, where naturalness is defined for participants in the following way: "a natural sentence is something that a native speaker of English would naturally and fluently say, following the implicit structural rules of English." As an example, participants are told that (32a) is natural, (32b) is "somewhat natural, because—even though it doesn't make sense (toothbrushes don't sleep)—it does follow the structural rules of English" and (32c) is unnatural.

(32) a. The baby seems to be sleeping.
b. The toothbrush seems to be sleeping.
c. The baby seems sleep to be.

Participants are then given three practice sentences similar to those in (32). Participants are given feedback on these practice sentences, which they have to rate correctly—rating the sentences like (32a)–(32b) above the midpoint, and the one like (32c) below the midpoint—in order to continue. Participants that did not correctly complete the practice sentences within two attempts were not allowed to continue.

4.2 Subexperiment 1.1: Manually generated sentences

Subexperiment 1.1 has two purposes: (i) to assess the extent to which experts can reliably generate natural and unnatural examples; and (ii) to produce a set of non-target examples beyond those discussed in Section 4.1 to be used in calibrating participants to the rating scale in Subexperiment 1.2.

4.2.1 Materials

The process for generating 124 manually-selected sentences is detailed in Section 3.2. Participants rated all sentences in a randomized order after a practice block of 3 questions.

4.2.2 Participants

We recruited 55 participants to rate our manually-generated sentences, all who passed our practice questions, with the goal of having each sentence rated by at least 5 annotators. All annotators were self-identified native English speakers located in the United States and recruited through the Prolific web platform. No annotator was allowed to participate in more than one of these studies. With IRB approval from our institutions, annotators were paid an average of $12/hour.

4.2.3 Selecting calibration sentences

We selected 50 of the 124 manually generated sentences to use as calibration sentences in Subexperiment 1.2. The point of these calibration sentences is to ensure that the responses participants provide in Subexperiment 1.2 are comparable to those that we observe in Subexperiment 1.1. Ensuring this comparability is crucial, since if we only provided participants with the automatically generated sentences, they may calibrate

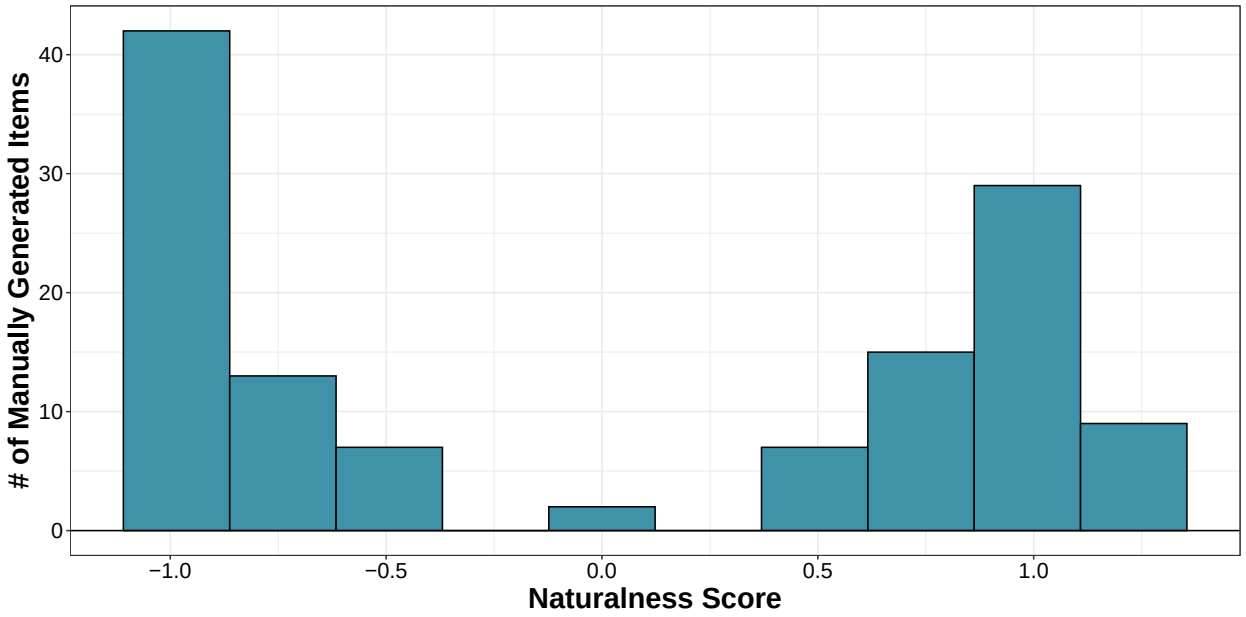

Figure 1: Distribution of naturalness scores for manually generated sentences.

to the variability in naturalness found among those sentences, making the ratings incomparable to the ratings for the manually generated sentences.

In selecting calibration sentences, we aim for a subset of sentences (i) in which the ratings are as uniformly distributed as possible; (ii) in which the mean rating is as close as possible to the mean across all sentences; (iii) that contains sentences headed by the same verb no more than twice; and (iv) for verbs found in two sentences, the difference in the naturalness ratings for that verb's sentences is at least 0.5 standard deviations. To satisfy these criteria, we first z-scored the naturalness ratings by participant in order to normalize for differences in scale use and took the mean of these z-scored ratings by sentence to derive a single naturalness score for each sentence. Figure 1 shows the distribution of naturalness scores. The bimodal distribution observed in Figure 1 is expected, given that we engineered sentences to be either natural or unnatural.

We then selected sentences for inclusion by (i) sorting sentences by their absolute distance to the mean naturalness score across sentences; (ii) moving through the sort, keeping sentences that move the mean of the included sentences toward the mean across all sentences and rejecting sentences that either (a) move the subset mean in the wrong direction; or (b) that would violate the fourth constraint described above.

Of this subset of 50 sentences, the eight sentences that were selected first in the procedure described above were reserved as the first sentences that participants saw in Subexperiment 1.2 (before any target sentences were shown and after the guided

questions and introduction). We refer to this set as the *initial calibration set*, used in a *calibration block*. The remainder were pseudorandomly interleaved with the target sentences in Subexperiment 1.2 in their specified order to ensure that participants remain calibrated over the course of the experiment.

4.3 Subexperiment 1.2: Automatically generated sentences

Although we technically had three different naturalness experiments, one per automatic generation method, we consider all the data together and treat the different methods as factors between subjects.

4.3.1 Materials

The process for automatically generating corpus-based sentences with Propbank senses, LM sentences with Propbank senses, and LM sentences with LM senses is detailed in Section 2. Since there were over 300 sentences generated per process, we used lists to allocate a reasonable number of sentences per participant. In order to generate lists of target sentences interwoven with calibration sentences, we first assigned each target sentence to a random list number (with a seed of 0) in ascending order such that there are less than 60 target sentences with the same list number. After splitting these sentences into their allocated list, we add calibration sentences in the order in which we specified in Section 4.2.3 for a total of 89 sentences. Since there are already 3 practice sentences, and each survey also had a block of 8 calibration sentences shown after the instructions and practice sentences but before the target sentences, this leads to a total of 100 sentences per survey/list. There were 7 lists generated for the survey with corpus-based sentences, 10 lists generated for the survey with LM sentences with Propbank senses, and 15 lists generated for the survey using LM sentences with LM senses.

4.3.2 Participants

We recruited 61 participants to rate corpus-generated sentences of which 60 participants passed the practice questions; 62 participants to rate sentences generated using the LM prompted with PropBank senses of which 61 participants passed the practice questions; and 86 participants to rate sentences generated using the LM prompted with LM senses of which 79 participants passed the practice questions.

Every sentence was rated by at least 5 annotators, all self-identified native English speakers located in the United States and recruited through the Prolific web platform.

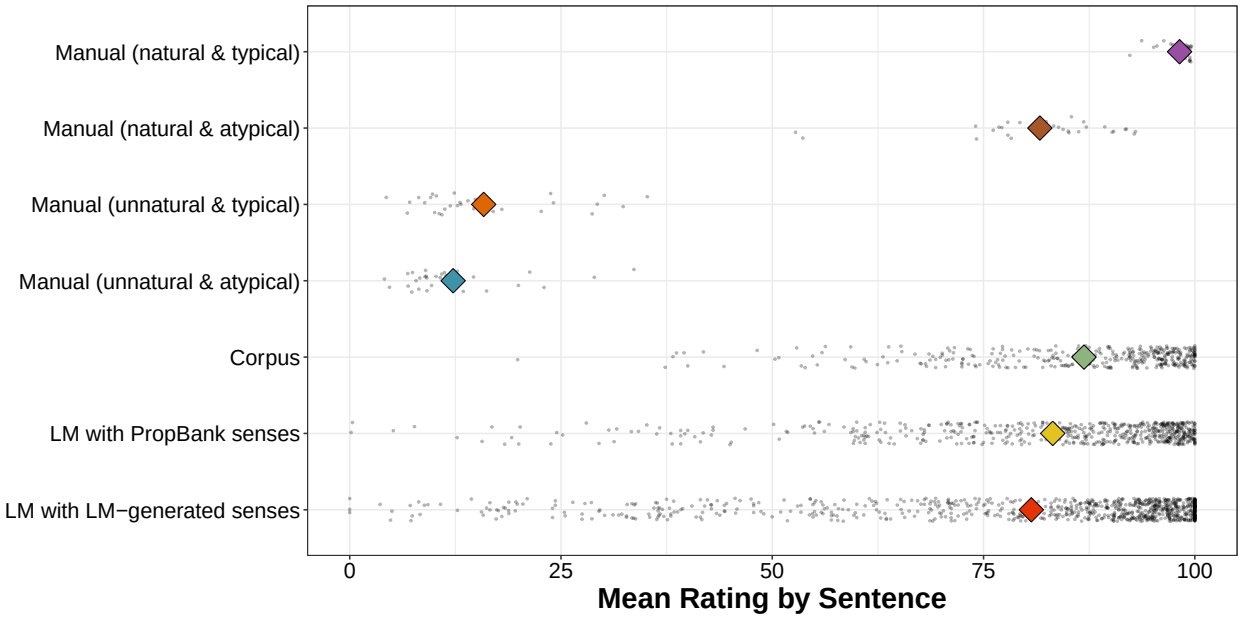

Figure 2: Mean naturalness rating for each sentence produced by each generation method. Each black point shows the mean rating of a sentence and large colored points show the mean of those means for each generation method.

No annotator was allowed to participate in more than one of these studies. With IRB approval from our institutions, annotators were paid an average of $12/hour.

4.4 Results

Figure 2 shows the mean naturalness ratings for each sentence from each generation method. Each black point shows the mean rating of a sentence and large colored points show the mean of those means. As expected, the manually generated examples that were both natural and typical are indeed rated as very natural. The examples that natural but atypical receive slightly lower ratings, potentially suggesting that naturalness is influenced by typicality. But this effect is nowhere near as strong as the effect of being manually engineered to be unnatural: all of the examples that were engineered to be unnatural are rated as such.

The automatically generated examples show naturalness ratings that tend to be about as good as—or slightly better than—the manually generated sentences constructed to be natural but atypical. We hypothesize that these ratings are a product of the automatically generated examples being less typical than the manually generated sentences that were engineered to be both natural and typical—a hypothesis we return to in Section 5.

	Post. mean	2.5%	97.5%	Post. p
Intercept	2.96	2.64	3.27	< 0.01
Manual (Natural & Atypical)	-2.15	-2.55	-1.79	< 0.01
Manual (Unnatural & Typical)	-4.54	-4.96	-4.11	< 0.01
Manual (Unnatural & Atypical)	-4.78	-5.18	-4.39	< 0.01
Corpus	-1.69	-2.04	-1.33	< 0.01
LM with manually-generated sense glosses	-1.89	-2.24	-1.55	< 0.01
LM with LM-generated sense glosses	-1.88	-2.21	-1.54	< 0.01

Table 7: Fixed effect coefficient estimates for naturalness experiment with *Manual (Natural & Typical)* as the reference-level in a dummy coding. The 2.5% and 97.5% columns give the lower and upper bound of the 95% credible interval, respectively, and the posterior p column gives the posterior probability that the coefficient has a sign different from the posterior mean. The posterior mean of the lower cutpoint of the ordered beta model is -2.0 (95% CrI = $[-2.05, -1.95]$) and the posterior mean of the mean of the upper cutpoints is 0.45 (95% CrI = [0.14, 0.75]).

4.5 Analysis

To assess the reliability of the differences observed in Figure 2, we fit an ordered beta mixed effects model (Kubinec, 2023) to the responses. This model had fixed effects for generation method—treating classes of manually generated sentences as separate generation methods—as well as by-participant, by-verb, by-sense, and by-sentence random intercepts. Table 7 shows the estimates for the fixed effect coefficients. All automatic generation methods produce reliably less natural examples than manual generation, though all such methods produce examples that are more natural than the manually generated natural, atypical examples: corpus (posterior $p > 0.99$), LM with PropBank senses (posterior $p = 0.95$), and LM with LM senses (posterior $p = 0.97$).

4.6 Discussion

We find that examples produced by the automated methods we consider are not as natural as our best manually generated sentences but that they are nonetheless quite natural—being rated as more natural, on average, than sentences we manually generated to be natural but atypical. We hypothesize that these ratings are a product of the automatically generated examples being less typical than the manually generated sentences that were engineered to be both natural and typical. We investigate this hypothesis in Section 5 but preliminarily conclude from the pattern observed in Experiment 1 that all of the automated generation methods we consider are safe to use

for generating examples for downstream annotation insofar as some small amount of degradation in naturalness is tolerable.

5 Experiment 2: Typicality

Like the naturalness experiment, our typicality experiment was separated into two subexperiments: one focused on the manually generated sentences (Section 5.2) and the other focused on the automatically generated sentences (Section 5.3).

5.1 Instructions and practice sentences

In both subexperiments, participants are asked to rate the typicality of the event described by a sentence on a continuous scale ranging from *very atypical* to *very typical*. They are instructed that "a typical situation is one in which individuals and their actions follow our normal expectations." They are told that (33a) describes a typical situation, while (33b) is less typical because "waitresses typically serve food instead of cooking it and salads are not cooked;" and (33c) is the least typical of all because "aliens are not typically associated with cooking and pencils are not things that are typically cooked."

(33) a. The chef cooked the meal.
b. The waitress cooked the salad.
c. The alien cooked the pencil.

Participants are then given three practice sentences similar to (33). Participants are given feedback on these practice sentences, which they have to rate correctly—rating the sentences like (33a)–(33b) above the midpoint, and the one like (33c) below the midpoint—in order to continue.

5.2 Subexperiment 2.1: Manually generated sentences

5.2.1 Materials

The process for generating manually-selected sentences is detailed in Section 2.1. Similar to Subexperiment 1.1, all sentences were shown to participants in a randomized order, after a practice block of 3 questions.

5.2.2 Participants

Similarly to Subexperiment 1.1, we recruited 55 participants from the same pool (although no participants participated in more than one of our subexperiments). They were compensated at the same rate.

5.2.3 Selecting calibration senteces

Our procedure for selecting calibration sentences for our future typicality subexperiments is directly analagous to the procedure described in Section 4.2.3.

5.3 Subexperiment 2.2: Automatically generated sentences

5.3.1 Materials

The procedure for generating the automatically generated sentences are detailed in Section 3. Similar to Subexperiment 1.2, since these processes yielded too many sentences to all be shown to participants in a single survey, we used the same lists generated in Section 4.3.1.

5.3.2 Participants

We recruited 60 participants to rate corpus-generated sentences of which all 60 participants passed the practice questions; 62 participants to rate sentences generated using the LM prompted with PropBank senses of which 60 participants passed the practice questions; and 79 participants to rate sentences generated using the LM prompted with LM senses of which all 79 participants passed the practice questions.

Every sentence was rated by at least 5 annotators, all self-identified native English speakers located in the United States and recruited through the Prolific web platform. No annotator was allowed to participate in more than one of these studies. With IRB approval from our institutions, annotators were paid an average of $12/hour.

5.4 Results

Figure 3 shows the mean typicality ratings for each sentence from each generation method. Each black point shows the mean rating of a sentence and large colored points show the mean of those means.

As expected, the manually generated sentences that were constructed to be both natural and typical are indeed rated as very typical. All other categories of manually generated sentences received very low typicality ratings on average—including the

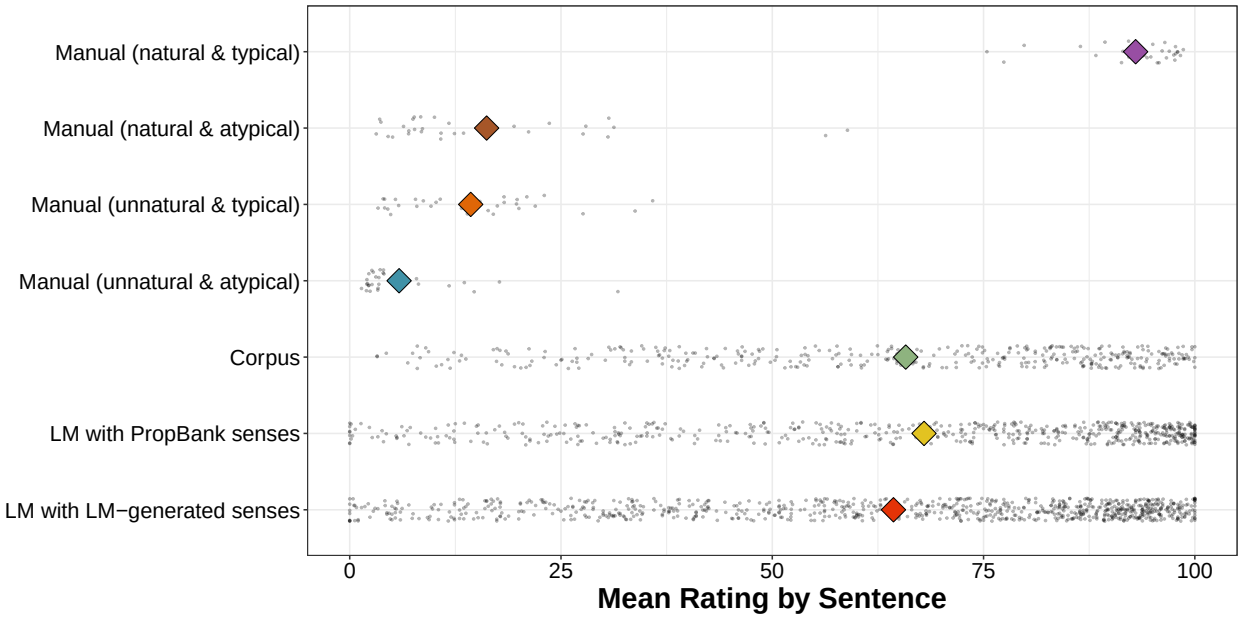

Figure 3: Mean typicality rating for each sentence from each source. Each black point shows the mean rating of a sentence and large colored points show the mean of those means for each generation method.

sentences that were constructed to be unnatural but typical. Thus, it appears that naturalness exerts a strong effect on typicality—with only natural sentences being candidates for typical event descriptions.

The automatically generated examples are rated as less typical than the manually generated sentences that are constructed to be natural and typical, though they are all rated to be substantially more typical than any of the other categories of manually generated sentences.

5.5 Analysis

To assess the reliability of the differences observed in Figure 3, we fit an ordered beta mixed effects model to the responses. This model had fixed effects for generation method—treating classes of manually generated sentences as separate generation methods—as well as by-participant, by-verb, by-sense, and by-sentence random intercepts. Table 8 shows the estimates for the fixed effect coefficients. Consistent with the pattern observed in Figure 3, all automatic generation methods produce reliably less typical event descriptions than manual generation, though all such methods produce examples that are more typical than the manually generated sentences that are constructed to be unnatural or atypical (all posterior ps < 0.01).

	Post. mean	2.5%	97.5%	Post. p
Intercept	1.85	1.49	2.21	< 0.01
Manual (Natural & Atypical)	-3.45	-3.85	-3.02	< 0.01
Manual (Unnatural & Typical)	-3.64	-4.08	-3.17	< 0.01
Manual (Unnatural & Atypical)	-4.35	-4.81	-3.86	< 0.01
Corpus	-1.41	-1.80	-1.06	< 0.01
LM with manually-generated sense glosses	-1.38	-1.75	-1.01	< 0.01
LM with LM-generated sense glosses	-1.52	-1.89	-1.16	< 0.01

Table 8: Fixed effect coefficient estimates for typicality experiment with *Manual (Natural & Typical)* as the reference-level in a dummy coding. The 2.5% and 97.5% columns give the lower and upper bound of the 95% credible interval, respectively, and the posterior p column gives the posterior probability that the coefficient has a sign different from the posterior mean. The posterior mean of the lower cutpoint of the ordered beta model is -1.86 (95% CrI = [-1.91, -1.82]) and the posterior mean of the mean of the upper cutpoints is 1.11 (95% CrI = [0.82, 1.45]).

5.6 Discussion

We find that event descriptions conveyed by sentences produced using automatic methods we consider are not as typical as our best manually generated sentences but that they are nonetheless substantially more typical than event descriptions manually constructed to be atypical. We conclude from this result that all of the automated generation methods we consider are safe to use for generating sentences for downstream annotation insofar as some small amount of degradation in typicality is tolerable.

In Section 4, we hypothesized that typicality may play a role in the degraded naturalness of the automatically generated sentences. To test this hypothesis, we derived a measure of typicality for each sentence by computing its average typicality rating z-scored by participant. We fit an ordered beta mixed effects model to the naturalness ratings reported on in Section 4 with this rating as a fixed effect as well as by-participant, by-verb, by-sense, and by-sentence random intercepts. We find a strong, reliably positive effect of typicality ($\beta = 1.51$, 95% CrI=[1.38, 1.63])—consistent with typicality exerting a strong influence on naturalness ratings. This result is consistent with the degradation in naturalness observed among the automatically generated sentences being driven mainly by typicality effects, rather than syntactic ill-formedness.

6 Experiment 3: Distinctiveness

Like the naturalness and typicality experiments, our distinctiveness experiment was separated into two subexperiments: one focused on the manually generated examples

(Section 6.2) and the other focused on the automatically generated examples (Section 6.3).

6.1 Instructions and practice sentences

Participants were presented with two sentences describing situations and asked "how similar or different the situations are compared to each other" on a slider scale ranging from *completely identical* to *extremely different*.[20] As an example, they were given the pair in (34) and told (34a) and (34b) are extremely different because the former involves the physical movement of an athlete participating in a race, while the latter involves the managing and leadership of a company.

(34) a. The athlete ran the race.
b. The manager ran the organization.

After receiving these instructions, participants were given two practice pairs that were manually generated to be obviously different and obviously similar and were required to rate these pairs above or below the midpoint of the scale, respectively, in order to progress to the main task.

6.2 Subexperiment 3.1: Manually generated sentences

Subexperiment 3.1 has two purposes: (i) to assess the extent to which experts can reliably generate same-sense and sense-distinct examples; and (ii) to produce a set of pairs of examples to be used in calibrating participants in our forthcoming distinctiveness experiments.

6.2.1 Materials

A second set of manual sentences were generated for the pair difference task, as described as the second set of calibration sentences in Section 3.2. In sum, for a *constant sentence*, two additional sentences were generated: one that uses the same sense of the verb as the constant sentence, and one that uses a different sense of the verb as the constant sentence. Although not all of the calibration verbs have more than one PropBank sense, each pair of verbs in this set was ensured to differ in some way (based

[20]This task is similar in design to Erk *et al.*'s (2009) Usage Similarity task, which she uses to explore graded meaning similarity of target words. The main difference is that we use a slider, rather than an ordinal scale.

on our own judgment)—e.g. (35a) involves social and/or temporal arrangement, while (35b) involves physical arrangement.

(35) a. The secretary arranged the meeting.
b. The baby arranged the blocks.

6.2.2 Participants

We recruited 56 participants through Prolific to rate our manually-generated pairs, all who passed our practice questions. All participants were English speakers from the United States who had not participated in one of our previous studies. With IRB approval from our institutions, annotators were paid an average of $12/hour.

6.2.3 Selecting calibration sentences

The calibration pairs for the pair difference task were generated using the same procedure as those used in the Naturalness and Typicality tasks. However, the 30 pairs of calibration sentences chosen for the norming study all used unique verbs. Of the 30 pairs, 6 were shown to every single participant (after the examples and before any target sentences).

6.3 Subexperiment 3.2: Automatically generated sentences

6.3.1 Materials

For each way of automatically generating sentences, and for each sense of each verb—as labeled by a sense-tagger on the corpus; as elicited from the LLM using PropBank glosses; as elicited from the LLM using the LLM's own senses—we found the two sentences with the highest sum of their z-scored ratings for naturalness and typicality from Experiments 1 and 2. These two sentences represent a *same-sense pair*. Then, using the top sentence for each verb-sense—the one with the highest sum of z-scored naturalness and typicality—we created *different-sense pairs*. So for a verb with n senses, there are $n + \binom{n}{k}$ pairs.

Since this process yields > 148 pairs of sentences per generation process, which is too many to be shown to a single participant, we implemented a similar list-making process to that described in Section 4.3.1. Specifically, we randomly assign each pair to a list number in ascending order such that no more than 60 unique pairs occur in the same list. Next, we padded each list with calibration sentences in their specified order until there were exactly 62 sentences per list. Since there are 2 practice questions

Source	Sentences	Sense gloss	Pair type
Corpus	The amateurs hit the road. The resolutioners hit the gym.	`hit.03` (go to, turn to) `hit.03` (go to, turn to)	Same
	The amateurs hit the road. The body hit the floor.	`hit.03` (go to, turn to) `hit.02` (reach, encounter)	Different
LM + PB senses	The ball hit the wall. The ball hit the net.	`hit.02` (reach, encounter) `hit.02` (reach, encounter)	Same
	The ball hit the wall. The road hit the lakefront.	`hit.02` (reach, encounter) `hit.03` (go to, turn to)	Different
LM + LM senses	The car hit the guardrail. The police hit the rioters.	use violence or force use violence or force	Same
	The car hit the guardrail. The trumpeter hit the highnotes.	use violence or force play a musical note	Different

Table 9: Examples of pairs of sentences used in the pair difference task.

and 6 sentences in the calibration block, this yields lists with 70 sentences each. We wanted the distinctiveness surveys to have list rating instances than the naturalness and typicality surveys because in pilots, we found that distinctiveness ratings take participants more time. In total, this process yields 3 lists generated for the survey with corpus-based sentences, 5 lists generated for the survey with LM sentences with Propbank senses, and 11 lists generated for the survey using LM sentences with LM senses.

6.3.2 Participants

We recruited 30 native English-speaking participants (of which all passed the practice questions) for the survey using LM sentences prompted with PropBank senses, 64 participants (of which 62 passed the practice questions) for the survey using LM sentences prompted with LM senses, and 20 participants (of which all passed the practice questions) for the survey using corpus-generated sentences on Prolific. No participants previously participated in any of our other surveys. Each participant was randomly shown a list from the list-making process delineated in Section 6.3.1.

6.4 Results

Figure 4 shows the mean distinctiveness ratings for each pair of sentences from each generation method. Each black point shows the mean rating of a sentence and large colored points show the mean of those means. As expected, the manually generated pairs constructed to instantiate the same sense were rated more similar on average

	Most different	**Least different**
Corpus	The state passed the resolution. The grandmother passed the aunt.	The client mailed the letter. The couple mailed the card.
LM + PropBank senses	The actor belted the tune. The father belted the child.	The soldier fired the rifle. The hunter fired the shotgun.
LM + LM senses	The boy beat the drum. The team beat the rivals.	The mother slapped the child. The parent slapped the teen.

Table 10: Most and least different pairs of non-filler sentences, according to our participants. Notably, all *most different* pairs contain different senses, while all *least different* pairs contain sentences using the same sense of the verb.

	Post. mean	**2.5%**	**97.5%**	**Post.** p
Intercept	-1.22	-1.61	-0.89	< 0.01
Different Sense	1.50	1.19	1.82	< 0.01
Corpus	0.19	-0.28	0.80	0.28
LM with PropBank senses	-0.36	-0.92	0.28	0.12
LM with LM senses	-0.07	-0.51	0.38	0.36
Different Sense $\times$ Corpus	-0.36	-0.73	0.00	0.03
Different Sense $\times$ LM with PropBank senses	0.05	-0.41	0.41	0.39
Different Sense $\times$ LM with LM senses	-0.37	-0.70	0.01	0.03

Table 11: Fixed effect coefficient estimates for distinctiveness experiment with *Same Sense* as the reference level for comparison type and *Manual* as the reference level for generation method in a dummy coding. The 2.5% and 97.5% columns give the lower and upper bound of the 95% credible interval, respectively, and the posterior p column gives the posterior probability that the coefficient has a sign different from the posterior mean. The posterior mean of the lower cutpoint of the ordered beta model is -2.51 (95% CrI = [-2.59, -2.43]) and the posterior mean of the mean of the upper cutpoints is 1.08 (95% CrI = [0.46, 1.82]).

than those constructed to instantiate two different senses. All other generation methods show smaller differences between the same sense pairs and the different sense pairs. In the case of the pairs constructed from corpus sentences, this difference appears to be due to the same sense pairs being more different than the corresponding manual sense pairs. In contrast, for the two LM-based methods, the different sense pairs tend to be less different than the manually generated different sense pairs.

6.5 Analysis

To assess the reliability of the differences observed in Figure 4, we fit an ordered beta mixed effects model to the responses. This model had fixed effects for genera-

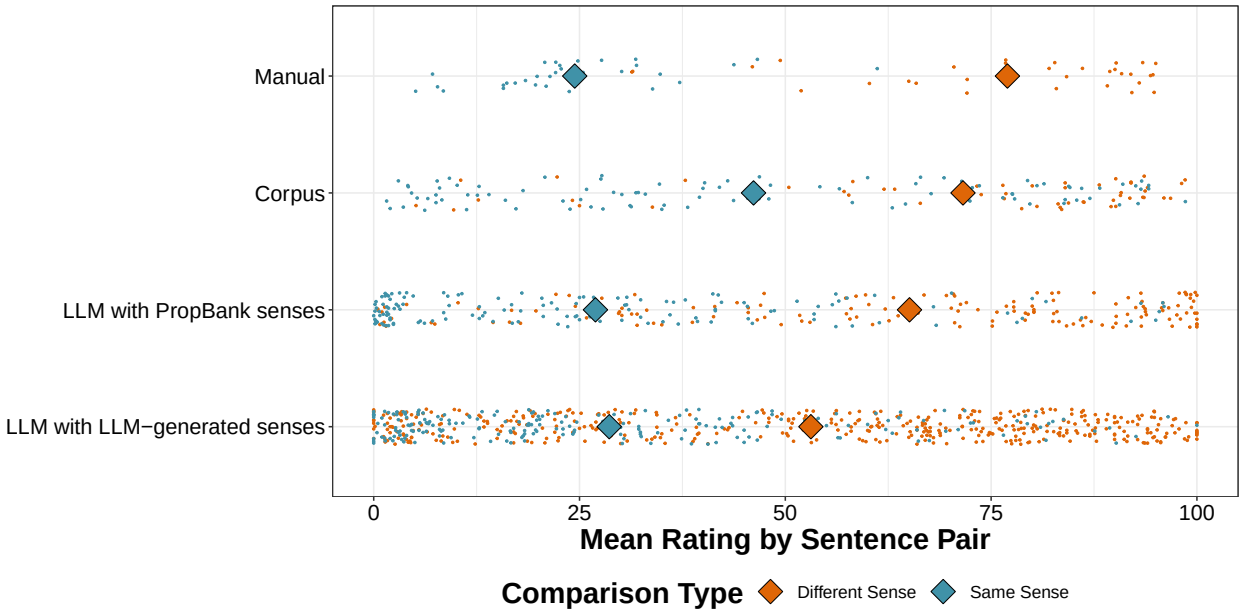

Figure 4: Mean difference rating for each sentence pair from each source. Each point shows the mean rating of a sentence pair and large colored points show the mean of those means for each relevant comparison type.

tion method—treating classes of manually generated sentences as separate generation methods—as well as by-participant, by-verb, by-sense pair, and by-sentence pair random intercepts. Table 11 shows the estimates for the fixed effect coefficients. Consistent with the pattern observed in Figure 4, the different sense pairs are rated as reliably more different than the same sense pairs for the manual sentences.

Interestingly, while the coefficient estimate for the corpus sentences is positive—as we would expect from Figure 4—the posterior probability that it is positive is somewhat low ($\sim$72%)—perhaps due to the high variability across same sense pairs observable in the Figure 4. The interaction with different sense has a high posterior probability of being negative, however, suggesting that the different sense corpus pairs were rated as reliably less different than the manually generated different sense pairs. We see a similar pattern for the LM-generated pairs conditioned on LM-generated senses.

In contrast, while the LM-generated different pairs conditioned on PropBank senses appear to be less different than the corresponding manually generated different sense pairs, the posterior probability of this is low ($\sim$61%). Indeed, it is more likely that the overall effect of LM-generation using PropBank senses on ratings is itself negative. This result, which likely arises from the fact that there are a substantial number of same sense pairs clustered around a rating of 0, suggests that both same sense and different sense pairs using this method are simply rated as more similar on the whole,

though the difference between same sense and different sense pairs is at about the same level as the manually generated sentences.

Nonetheless, even though the automated methods produce different sense pairs that are less different than the manually generated different sense pairs, all automated methods produce different sense pairs that are reliably more different than the maually generated same sense pairs (posterior ps < 0.01).

6.6 Discussion

We find that event descriptions produced by the automatic methods we consider are not as distinctive as our best manually generated sentences but that they are nonetheless substantially more distinctive than event descriptions manually constructed to use the same sense. We conclude from this result that all of the automated generation methods we consider are safe to use for generating sentences for downstream annotation insofar as some small amount of degradation in distinctiveness is tolerable.

7 Conclusion

In three experiments, we have demonstrated that automated methods for generating linguistic expressions for downstream annotation and analysis can generate reliably natural, typical, and distinctive event descriptions across different senses of verbs—though all automated methods we investigated yield less natural, typical, and distinctive linguistic expressions than manual generation can—summarized in Table 12. We draw two main conclusions from these findings.

First, we conclude that, while the automated methods can be used to generate linguistic expressions of reasonably good quality, they should largely be reserved for approaches to data collection and analysis (i) in which it is simply not feasible to manually generate all linguistic expressions of interest; and (ii) that can be made robust to degradation in linguistic expressions' naturalness, typicality, and distinctiveness. Effectively, we submit that automated generation methods are a good option for lexical semantic research wherein large portion of the lexicon are being studied—e.g. if one were interested in drawing sweeping generalizations about all clause-embedding predicates and were worried about known methodological issues that are known to arise from poor sampling methodologies (see White, 2021)—but that there is little reason to use them if smaller portions of the lexicon are under investigation. Second, we conclude that LM-based methods yield linguistic expressions of sufficiently comparable quality to those yielded by corpus-based methods—at least under the sorts of heavy

	Quality	Effort	Efficiency
Manual	High	High	Low
Corpus	Medium	Low	Medium
LLM	Medium	Low	High

Table 12: Final summary of sampling methods for linguistic stimuli (i.e., sentences).

constraints we impose in this paper—that they are preferable (all else being equal) to corpus-based methods for their efficiency.

These results raise a number of potentially interesting questions for future research. First, how well do the automated methods we consider—and the LM-based methods in particular—generalize to more complex syntactic constraints? We considered relatively strict constraints, resulting in a relatively simple syntactic context for the verbs of interest. Do the automated methods we consider gracefully scale to constraints that result in more complex syntactic structures?

Second, how well do the automated methods we consider generalize to more complex semantic or pragmatic constraints? For instance, if we are interested in generating multisentence contexts, how far can a combination of prompting and constrained decoding from an LM take us in the absence of nontrivial modifications to that LM?

Finally, to what extent can we improve the outputs of the automated methods we consider in an efficient way? For instance, is it possible to deploy efficient post-editing of automatically generated linguistic expressions—as in, e.g., Green *et al.* 2013, i.a.—to bring the quality of those expressions to the level of fully manually generated expressions in a way that reduces the overall human effort required by full manual generation?

References

An, Hannah, & White, Aaron. 2020. The lexical and grammatical sources of neg-raising inferences. *Proceedings of the Society for Computation in Linguistics*, **3**(1), 220–233.

Baumgartner, Jason, Zannettou, Savvas, Keegan, Brian, Squire, Megan, & Blackburn, Jeremy. 2020. The PushShift Reddit Dataset. *Pages 830–839 of: Proceedings of the International AAAI (Association for the Advancement of Artificial Intelligence) Conference on Web and Social Media*, vol. 14.

Brown, Tom, Mann, Benjamin, Ryder, Nick, Subbiah, Melanie, Kaplan, Jared D,

Dhariwal, Prafulla, Neelakantan, Arvind, Shyam, Pranav, Sastry, Girish, Askell, Amanda, Agarwal, Sandhini, Herbert-Voss, Ariel, Krueger, Gretchen, Henighan, Tom, Child, Rewon, Ramesh, Aditya, Ziegler, Daniel, Wu, Jeffrey, Winter, Clemens, Hesse, Chris, Chen, Mark, Sigler, Eric, Litwin, Mateusz, Gray, Scott, Chess, Benjamin, Clark, Jack, Berner, Christopher, McCandlish, Sam, Radford, Alec, Sutskever, Ilya, & Amodei, Dario. 2020. Language Models are Few-Shot Learners. *Pages 1877–1901 of: Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc.

Christiano, Paul F., Leike, Jan, Brown, Tom B., Martic, Miljan, Legg, Shane, & Amodei, Dario. 2017. Deep reinforcement learning from human preferences. *Pages 4302–4310 of: Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Red Hook, NY, USA: Curran Associates Inc.

Dathathri, Sumanth, Madotto, Andrea, Lan, Janice, Hung, Jane, Frank, Eric, Molino, Piero, Yosinski, Jason, & Liu, Rosanne. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. *In: International Conference on Learning Representations*.

Deutsch, Daniel, Upadhyay, Shyam, & Roth, Dan. 2019. A General-Purpose Algorithm for Constrained Sequential Inference. *Pages 482–492 of:* Bansal, Mohit, & Villavicencio, Aline (eds), *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, & Toutanova, Kristina. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Pages 4171–4186 of:* Burstein, Jill, Doran, Christy, & Solorio, Thamar (eds), *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)*. Minneapolis, Minnesota: Association for Computational Linguistics.

Earley, Jay. 1970. An efficient context-free parsing algorithm. *Communications of the ACM*, **13**(2), 94–102.

Erk, Katrin, McCarthy, Diana, & Gaylord, Nicholas. 2009. Investigations on Word Senses and Word Usages. *Pages 10–18 of: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore: Association for Computational Linguistics.

Fillmore, Charles J. 1970. The grammar of hitting and breaking. *Pages 120–33 of:* Jacobs, Roderick, & Rosenbaum, Peter (eds), *Readings in English transformational grammar*. Washington: Georgetown University Press.

Green, Spence, Heer, Jeffrey, & Manning, Christopher D. 2013. The efficacy of human post-editing for language translation. *Pages 439–448 of: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '13. New York, NY, USA: Association for Computing Machinery.

Hale, John. 2001. A Probabilistic Earley Parser as a Psycholinguistic Model. *In: Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Holtzman, Ari, Buys, Jan, Forbes, Maxwell, Bosselut, Antoine, Golub, David, & Choi, Yejin. 2018. Learning to Write with Cooperative Discriminators. *Pages 1638–1649 of:* Gurevych, Iryna, & Miyao, Yusuke (eds), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics.

Holtzman, Ari, Buys, Jan, Du, Li, Forbes, Maxwell, & Choi, Yejin. 2019 (Sept.). The Curious Case of Neural Text Degeneration.

Honnibal, Matthew, & Montani, Ines. 2017. *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*.

Hovy, Eduard, Marcus, Mitchell, Palmer, Martha, Ramshaw, Lance, & Weischedel, Ralph. 2006. OntoNotes: The 90% Solution. *Pages 57–60 of:* Moore, Robert C., Bilmes, Jeff, Chu-Carroll, Jennifer, & Sanderson, Mark (eds), *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. New York City, USA: Association for Computational Linguistics.

Kallmeyer, Laura, & Maier, Wolfgang. 2013. Data-Driven Parsing using Probabilistic Linear Context-Free Rewriting Systems. *Computational Linguistics*, **39**(1), 87–119. Place: Cambridge, MA Publisher: MIT Press.

Kane, Benjamin, Gantt, Will, & White, Aaron Steven. 2022. Intensional Gaps: Relating veridicality, factivity, doxasticity, bouleticity, and neg-raising. *Semantics and Linguistic Theory*, **31**(Jan.), 570–605. Number: 0.

Kato, Yuki, Seki, Hiroyuki, & Kasami, Tadao. 2006. Stochastic Multiple Context-Free Grammar for RNA Pseudoknot Modeling. *Pages 57–64 of:* Becker, Tilman, &

Kallmeyer, Laura (eds), *Proceedings of the Eighth International Workshop on Tree Adjoining Grammar and Related Formalisms*. Sydney, Australia: Association for Computational Linguistics.

Katz, Graham. 2019. Semantics in corpus linguistics. *Pages 409–443 of:* Heusinger, Klaus von, Maienborn, Claudia, & Portner, Paul (eds), *Semantics - Typology, Diachrony and Processing*. De Gruyter Mouton.

Kipper, Karin, Korhonen, Anna, Ryant, Neville, & Palmer, Martha. 2006. Extending VerbNet with novel verb classes. *In: Proceedings of 5th International Conference on Language Resources and Evaluation*, vol. 2006.

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

Kubinec, Robert. 2023. Ordered Beta Regression: A Parsimonious, Well-Fitting Model for Continuous Data with Lower and Upper Bounds. *Political Analysis*, **31**(4), 519–536.

Levin, Beth. 1993. *English Verb Classes and Alternations A Preliminary Investigation*. Chicago and London: University of Chicago Press.

Levy, Roger. 2008. Expectation-based syntactic comprehension. *Cognition*, **106**(3), 1126–1177.

Liu, Yinhan, Ott, Myle, Goyal, Naman, Du, Jingfei, Joshi, Mandar, Chen, Danqi, Levy, Omer, Lewis, Mike, Zettlemoyer, Luke, & Stoyanov, Veselin. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*, July. arXiv: 1907.11692.

Moon, Ellise, & White, Aaron Steven. 2020. The source of nonfinite temporal interpretation. *Pages 11–24 of:* Mariam Asatryan, Yixiao Song, & Ayana Whitmal (eds), *Proceedings of the 50th Annual Meeting of the North East Linguistic Society*, vol. 3. Amherst, MA: GLSA Publications.

Orlando, Riccardo. 2020. *transformer-srl*.

Ouyang, Long, Wu, Jeff, Jiang, Xu, Almeida, Diogo, Wainwright, Carroll L., Mishkin, Pamela, Zhang, Chong, Agarwal, Sandhini, Slama, Katarina, Ray, Alex, Schulman, John, Hilton, Jacob, Kelton, Fraser, Miller, Luke, Simens, Maddie, Askell, Amanda, Welinder, Peter, Christiano, Paul, Leike, Jan, & Lowe, Ryan. 2022. Training language models to follow instructions with human feedback. *Pages 27730–27744 of:*

Proceedings of the 36th International Conference on Neural Information Processing Systems. NIPS '22. Red Hook, NY, USA: Curran Associates Inc.

Palmer, Martha, Gildea, Daniel, & Kingsbury, Paul. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, **31**(1), 71–106.

Petersen, Erika, & Potts, Christopher. 2023. Lexical Semantics with Large Language Models: A Case Study of English "break". *Pages 490–511 of: Findings of the Association for Computational Linguistics: EACL 2023*. Dubrovnik, Croatia: Association for Computational Linguistics.

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

Pradhan, Sameer, Moschitti, Alessandro, Xue, Nianwen, Uryupina, Olga, & Zhang, Yuchen. 2012. CoNLL-2012 Shared Task: Modeling Multilingual Unrestricted Coreference in OntoNotes. *Pages 1–40 of:* Pradhan, Sameer, Moschitti, Alessandro, & Xue, Nianwen (eds), *Joint Conference on EMNLP and CoNLL - Shared Task*. Jeju Island, Korea: Association for Computational Linguistics.

Radford, Alec, Wu, Jeffrey, Child, Rewon, Luan, David, Amodei, Dario, & Sutskever, Ilya. 2018. *Language models are unsupervised multitask learners*. Technical Report. OpenAI.

Reisinger, Drew, Rudinger, Rachel, Ferraro, Francis, Harman, Craig, Rawlins, Kyle, & Van Durme, Benjamin. 2015. Semantic Proto-Roles. *Transactions of the Association for Computational Linguistics*, **3**, 475–488.

Shi, Peng, & Lin, Jimmy. 2019 (Apr.). *Simple BERT Models for Relation Extraction and Semantic Role Labeling*. arXiv:1904.05255 [cs].

Shin, Richard, Lin, Christopher, Thomson, Sam, Chen, Charles, Roy, Subhro, Platanios, Emmanouil Antonios, Pauls, Adam, Klein, Dan, Eisner, Jason, & Van Durme, Benjamin. 2021. Constrained Language Models Yield Few-Shot Semantic Parsers. *Pages 7699–7715 of:* Moens, Marie-Francine, Huang, Xuanjing, Specia, Lucia, & Yih, Scott Wen-tau (eds), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Smith, Noah A. 2011. *Linguistic Structure Prediction*. 1st edn. Morgan & Claypool Publishers.

Stiennon, Nisan, Ouyang, Long, Wu, Jeff, Ziegler, Daniel M., Lowe, Ryan, Voss, Chelsea, Radford, Alec, Amodei, Dario, & Christiano, Paul. 2020. Learning to summarize from human feedback. *Pages 3008–3021 of: Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Red Hook, NY, USA: Curran Associates Inc.

Stolcke, Andreas. 1995. An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities. *Computational Linguistics*, **21**(2), 165–201. Place: Cambridge, MA Publisher: MIT Press.

Touvron, Hugo, Martin, Louis, Stone, Kevin, Albert, Peter, Almahairi, Amjad, Babaei, Yasmine, Bashlykov, Nikolay, Batra, Soumya, Bhargava, Prajjwal, Bhosale, Shruti, Bikel, Dan, Blecher, Lukas, Ferrer, Cristian Canton, Chen, Moya, Cucurull, Guillem, Esiobu, David, Fernandes, Jude, Fu, Jeremy, Fu, Wenyin, Fuller, Brian, Gao, Cynthia, Goswami, Vedanuj, Goyal, Naman, Hartshorn, Anthony, Hosseini, Saghar, Hou, Rui, Inan, Hakan, Kardas, Marcin, Kerkez, Viktor, Khabsa, Madian, Kloumann, Isabel, Korenev, Artem, Koura, Punit Singh, Lachaux, Marie-Anne, Lavril, Thibaut, Lee, Jenya, Liskovich, Diana, Lu, Yinghai, Mao, Yuning, Martinet, Xavier, Mihaylov, Todor, Mishra, Pushkar, Molybog, Igor, Nie, Yixin, Poulton, Andrew, Reizenstein, Jeremy, Rungta, Rashi, Saladi, Kalyan, Schelten, Alan, Silva, Ruan, Smith, Eric Michael, Subramanian, Ranjan, Tan, Xiaoqing Ellen, Tang, Binh, Taylor, Ross, Williams, Adina, Kuan, Jian Xiang, Xu, Puxin, Yan, Zheng, Zarov, Iliyan, Zhang, Yuchen, Fan, Angela, Kambadur, Melanie, Narang, Sharan, Rodriguez, Aurelien, Stojnic, Robert, Edunov, Sergey, & Scialom, Thomas. 2023 (July). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. arXiv:2307.09288 [cs].

White, Aaron Steven. 2021. On believing and hoping whether. *Semantics and Pragmatics*, **14**(6), 1–18.

White, Aaron Steven, & Rawlins, Kyle. 2016. A computational model of S-selection. *Semantics and Linguistic Theory*, **26**, 641–663.

White, Aaron Steven, & Rawlins, Kyle. 2018. The role of veridicality and factivity in clause selection. *Page to appear of: Proceedings of the 48th Annual Meeting of the North East Linguistic Society*. Amherst, MA: GLSA Publications.

White, Aaron Steven, & Rawlins, Kyle. 2020. Frequency, acceptability, and selection: A case study of clause-embedding. *Glossa: a journal of general linguistics*, **5**(1), 105. Number: 1 Publisher: Ubiquity Press.

White, Aaron Steven, Reisinger, Drew, Sakaguchi, Keisuke, Vieira, Tim, Zhang, Sheng, Rudinger, Rachel, Rawlins, Kyle, & Van Durme, Benjamin. 2016. Universal Decompositional Semantics on Universal Dependencies. *Pages 1713–1723 of: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics.

White, Aaron Steven, Stengel-Eskin, Elias, Vashishtha, Siddharth, Govindarajan, Venkata Subrahmanyan, Reisinger, Dee Ann, Vieira, Tim, Sakaguchi, Keisuke, Zhang, Sheng, Ferraro, Francis, Rudinger, Rachel, Rawlins, Kyle, & Van Durme, Benjamin. 2020. The Universal Decompositional Semantics Dataset and Decomp Toolkit. *Pages 5698–5707 of: Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association.

Yang, Kevin, & Klein, Dan. 2021. FUDGE: Controlled Text Generation With Future Discriminators. *Pages 3511–3535 of:* Toutanova, Kristina, Rumshisky, Anna, Zettlemoyer, Luke, Hakkani-Tur, Dilek, Beltagy, Iz, Bethard, Steven, Cotterell, Ryan, Chakraborty, Tanmoy, & Zhou, Yichao (eds), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics.

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

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