Empirical Sufficiency Lower Bounds for Language Modeling with Locally-Bootstrapped Semantic Structures

Jakob Prange and **Emmanuele Chersoni**

Department of Chinese and Bilingual Studies, The Hong Kong Polytechnic University
11 Yuk Choi Road, Hung Hom, Kowloon, Hong Kong (China)
`first.last@polyu.edu.hk`

Abstract

In this work we build upon negative results from an attempt at language modeling with predicted semantic structure, in order to establish empirical lower bounds on what *could* have made the attempt successful.[1] More specifically, we design a concise binary vector representation of semantic structure at the lexical level and evaluate in-depth how good an incremental tagger needs to be in order to achieve better-than-baseline performance with an end-to-end *semantic-bootstrapping language model*. We envision such a system as consisting of a (pretrained) sequential-neural component and a hierarchical-symbolic component working together to generate text with low surprisal and high linguistic interpretability. We find that (a) dimensionality of the semantic vector representation *can* be dramatically reduced without losing its main advantages and (b) lower bounds on prediction quality cannot be established via a single score alone, but need to take the *distributions* of signal and noise into account.

1 Introduction

It is well-established by now that large pretrained Transformer language models (LMs) can obtain detectable knowledge about linguistic structure from raw text distributions (Jawahar et al., 2019; Tenney et al., 2019a, *inter alia*), thus continuing a long line of research in collecting solid empirical evidence for the Distributional Hypothesis (Harris, 1954; Firth, 1957). This is often presented in stark contrast to symbolic linguistic theories and representations, which put more emphasis on higher-level structural principles. In practice, purely neural models have achieved groundbreaking performances in a wide range of NLP tasks (Devlin et al., 2019; Brown et al., 2020) in a much more scalable manner than seems to be possible with symbolic ones. Still, theoretical linguistic questions
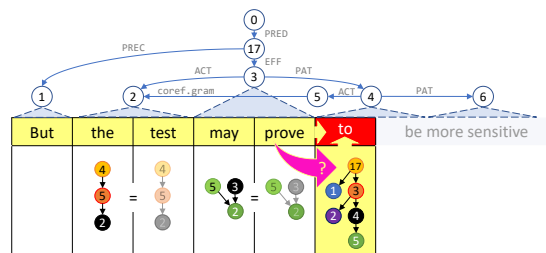

Figure 1: Example of incremental semantic graph slices obtained from a PTG graph and information flow in a (hypothetical) semantic-bootstrapping LM. In this example, the dark-red-shaded token 'to' is the current LM target; the light-yellow-shaded cells to the left and below directly influence the LM decision as in P+22; and the pink arrow marked with **?** stands for the intermediate slice prediction step, for which we want to establish sufficiency lower bounds.

about the relationship between neural implementation and higher-level symbolic patterns are far from being answered definitively. A common criticism of purely distributional models is that they generally *lack grounding*, because they do not have access to the external world, while *meaning* is inherently a relation between a linguistic form and a communicative intent about something external to language (Bender and Koller, 2020; Trott et al., 2020; Merrill et al., 2021; Lenci, 2023).[2]

We aim to contribute to this discussion by building upon results by Prange et al. (2022, henceforth P+22), who found that incremental LM perplexity can be significantly improved by providing hierarchical semantic structure as an additional token-level input (fig. 1 and §2.1). Indeed, the integration of symbolic and distributional approaches has long been seen as a possible and necessary step towards the full legitimacy of Distributional Semantic Models (DSMs) as models of meaning (Boleda and Herbelot, 2016; Emerson, 2020), and there is recently

---

[1] Our experimental code is available at https://github.com/jakpra/SufficiencyLowerBounds.

[2] But see Abdou et al. (2021) for a more optimistic view, backed by empirical results.

more and more evidence supporting the benefits of hybrid neuro-symbolic models (e.g., Li and Srikumar, 2019; Li and Rush, 2020), especially for compositional and long-tail generalization (Weißenhorn et al., 2022; Prange et al., 2021) and interpretability (Opitz and Frank, 2022).

P+22's results seem to suggest that at least some aspects of symbolic semantic structure may not be contained in the incremental LM's representation—i.e., that these aspects might constitute an instance of grounding, which is helpful for language understanding, but not fully learnable from text alone. Alternatively, we consider the possibility that the crucial semantic information *could* be learned, extracted, or induced to a sufficient extent, if only explicit supervision were provided at training time. The notion of *sufficiency*, in our case, relates to the potential of improving over a baseline LM (§3). This paints a grand vision of semantic bootstrapping, i.e., a scenario in which the LM first commits to local semantic structure based on the revealed sentence prefix (pink ? arrow in fig. 1) and then uses its prediction to reduce the next token's perplexity. The work by P+22 established *upper bounds* by using an oracle setup where rich semantic structure inputs are known to be correct, not only during training but also at test time. As the main contribution of this work, assuming the local semantic bootstrapping scenario is feasible at all, we look instead for *lower bounds* on what would constitute *sufficient substance and accuracy* in predicted semantic structure for such an improved end-to-end neuro-symbolic LM.

Concretely, we conduct two analyses: First, we make P+22's original formulation of semantic graph slices (SGS) more parsimonious (§5). We extract binary vectors (B-SGS) representing only bare-bones (unlabeled and unanchored) structural relations (§5.2) and find that they are sufficient for improving LM perplexity over a strong baseline in the oracle setting (§5.3). Second, we measure how the language modeling benefits of B-SGS are affected by increasing levels of noise, aiming to emulate various imperfect taggers (§6). Interestingly, a comparison of two different shuffling mechanisms (§6.2) as well as a simple pilot tagger (§6.1) reveals that *how errors are distributed* throughout the data is much more important than overall error rate. Based on our observations, we establish sufficiency lower bounds of B-SGS for use in a semantic bootstrapping LM. We begin by providing

the reader with relevant background information from the literature (§2), defining concisely what we mean by *sufficiency lower bounds* (§3), and describing our data set and general experimental setup (§4). Finally, we discuss our findings and limitations within the bigger picture of ongoing research directions (§7).

2 Background

2.1 Language Modeling with Linguistic Graph Slices

P+22 proposed a type of ensemble language model, consisting of a pretrained Transformer and a neural encoder of symbolic linguistic structure, both jointly predicting the next token in a sentence, given the revealed prefix. They extract token-level "slices" from sentence-level graphs.

An incremental linguistic graph **slice** is defined as a connected subgraph minimally including a node directly anchored in the target token (or a preceding token if no such node is available) and extending vertically to include parents, grandparents, and children, horizontally to include left siblings, and diagonally to include children's parents ("co-parents") and parents' siblings ("aunts"). This is illustrated in fig. 1: The original sentence-level graph is shown above the sentence, and extracted token-level slices are shown below. Slices are then **encoded** as fixed-length vectors, including both edge label information and token anchor information. Out of two encoding methods, R-GCN (Schlichtkrull et al., 2018) and a simple concatenation- and averaging-based one, the latter is much faster at roughly equal model size and roughly equal LM quality, so we choose it in our experiments. In essence, the embeddings of all preceding tokens related to the target token in one of the structural ways listed above (parents, siblings, etc), as well as their one-hot-encoded edge labels, are concatenated in a specific pre-defined order. If there are multiple instances of a given relation, or multi-token anchors, their vector representations are averaged. Missing relations are zero-padded. The final slice vector is fed through a simple feed-forward encoder in order to compute logits over the vocabulary, which are finally added to the LM's logits before softmax normalization. The resulting **distribution** is used to compute the loss (during training) or predict the next token (at test time).

In their study, P+22 compared linguistic representations of several different flavors, including

syntactic and dependency frameworks. Here we focus on two semantic frameworks, PTG and EDS (§4.1), which structurally go beyond bilexical dependencies, and thus we use the term *semantic graph slice* (SGS). We further extend P+22's work by explicitly comparing their oracle setup against several versions of SGS with varying degrees of richness and correctness, stemming from either signal reduction (§5), automatic prediction (§6.1), or controlled noise induction (§6.2).

2.2 Related Work

Linguistic Analyses of LMs. A large number of studies in the LM literature has been dedicated to the analysis of the linguistic knowledge they encode. A common methodology employs *probing tasks*, where a simple model is asked to solve a task requiring linguistic knowledge using a representation derived from a LM with little or no specific linguistic supervision. If the model is successful, we then can infer that the LM encodes that knowledge (see Linzen et al., 2016; Tenney et al., 2019a,b; Hewitt and Liang, 2019; Liu et al., 2019; Wu et al., 2020; Chersoni et al., 2021; Geiger et al., 2021, *inter alia*). Probes can be particularly insightful when applied *contrastively* to sets of minimal sentence pairs that differ in their grammatical acceptability (Warstadt et al., 2020; Hu et al., 2020; Kim et al., 2019). Our approach of treating semantic structure as an input rather than an output of a neural LM is orthogonal to probing, but can similarly be used for inferences about what kind of knowledge is (not) already encoded in the baseline model. Recently, an interpretability method based on contrastive *explanations* (Jacovi et al., 2021) has been proposed to explain LM predictions on sets of minimal sentence pairs that differ in their grammatical acceptability, showing that the salient tokens for the LM preference of the correct form are quite well aligned with human knowledge of grammatical phenomena (Yin and Neubig, 2022).

Incremental Supertagging and Parsing. Predicting linguistic structure incrementally has been explored especially in the context of strongly-formulated lexico-syntactic grammar formalisms like CCG, in the form of incremental supertagging (Hassan et al., 2009; Ambati et al., 2015; Stanojević and Steedman, 2019, 2020). Having word-level structural categories built in to the formalism has many advantages for both modeling efficiency and linguistic interpretability. But also Penn Treebank-

style constituency syntax trees can be parsed incrementally using, e.g., language model grammars (Sartran et al., 2022; Dyer et al., 2016) or word-level beam search (Stern et al., 2017). Finally, another line of work aims to backpropagate linguistic knowledge into the LM itself by optimizing incremental structure prediction as an auxiliary objective (Qian et al., 2021; Glavaš and Vulić, 2021; Kitaev et al., 2022).

Model Explanations and Cognitive Predictions using Linguistic Symbols. Hale et al. (2018) proposed a method relying on probabilistic generative grammars (Dyer et al., 2016) and word-synchronous beam search that allows to extract predictive metrics of processing difficulty, such as surprisal and entropy. The authors showed that, using such metrics as predictors in a regression model, it was possible to predict the amplitude effects of several components of naturalistic EEG. Ek et al. (2019) enhance a LSTM-based LM with syntactic, semantic tags and dependency tree depth features, and reported that the additional linguistic knowledge did not increase the correlation with human ratings in a sentence acceptability task, although syntactic tags and dependency tree depth were helpful for lowering perplexity. Stanojević et al. (2021) used CCG-based predictors to improve a regression model of fMRI time course in six different brain regions, over and above predictors obtained with a simple context-free phrase structure grammar. Finally, Opitz and Frank (2022) presented a technique to partition the BERT sentence embeddings into different sub-embeddings, each one covering meaningful semantic aspects of sentences as represented in the Abstract Meaning Representations (AMR) framework. Experiments on zero-shot sentence and argument similarity tasks proved that the approach maintains a high-level of correlation with human judgements, while making the sentence embeddings interpretable.

3 Sufficiency Lower Bounds

We introduce the concept of *sufficiency*[3] *lower bounds* on the strength of a data signal ξ in order for a system Σ, which takes ξ as an input, to reach a certain performance threshold θ. In this work, the system Σ is a neuro-symbolic LM as proposed

---

[3]We do not consider *necessity* lower bounds here. I.e., we do not say that data signals of worse substance than our lower bounds cannot be sufficient. We say that distributions of at least lower-bound quality are probably sufficient.

by P+22 (§2.1), ξ is an SGS vector representation (§5) for each (sub)word token in a text corpus D, and θ is the baseline LM performance (measured as surprisal, §4.3). Establishing such bounds is important because ξ's richness may need to be reduced in one way or another—either by theoretical design (because small, simple representations and models are desirable; §5), or by practical necessity (due to unavoidable noise in predicting ξ; §6). A main takeaway from our exploratory study is that it is important to identify (i.e., define or measure) candidate bounds in a way that considers the signal's configuration as a whole, rather than focusing on a single aggregate metric.[4] Approached empirically, this involves computing (multivariate) distributions over ξ as instantiated in a data set D, such that when the system Σ is run on D, the quality of its output is at least θ (i.e., it outperforms a baseline). Simply put, if the signal ξ surpasses the sufficiency lower bound in D, it will likely enable the system Σ to reach performance θ or better.

4 Experimental Setup

4.1 Data

We use the jointly-annotated corpus of the cross-framework meaning representation parsing (MRP) shared tasks (Oepen et al., 2019, 2020), which consists of large parts of the English Wall Street Journal corpus. In particular, we examine two symbolic-structured linguistic representation frameworks, Prague Tectogrammatical Graphs (**PTG**; Sgall et al., 1986; Böhmová et al., 2003; Hajič et al., 2012) and Elementary Dependency Structures (**EDS**; Oepen and Lønning, 2006; Flickinger, 2000; Copestake et al., 2005), each of them focusing on different aspects of semantic predicate-argument structure. EDS derives from Minimal Recursion Semantics (MRS) and thus rather explicitly encodes nominal/referring expressions due to MRS' foundation in variable binding. PTG, on the other hand, is somewhat more guided by syntax and (case-)semantic roles. We use the same training split as P+22, but deviate slightly in using only the first 500 sentences of their development set and reporting most of our results and analyses on this subset. This is because we are reporting incremental results and wish to reserve substantial

---

[4]This somewhat circular-looking reasoning warrants full disclosure: We were already proponents of holistic, detailed evaluations over single-number benchmarks before this study, but were still surprised by most of our results, particularly the contrast between §6.1 and §6.2.

unseen data for unbiased full evaluation in future work. For comparison, we report a limited amount of aggregate scores over the test set in table 2.

4.2 Model Implementation

Our models (see §2.1 for a conceptual overview) and experiments are implemented in Python, building on P+22's codebase.[5] In addition to standard neural language modeling libraries used therein (PyTorch, huggingface), we also leverage the Pyro-PPL library (Bingham et al., 2018) to implement the variational autoencoder (§6.1).

We follow P+22 in using GPT-2 (Radford et al., 2019, 124M parameters) as the pretrained incremental language model and a simple multilayer perceptron (MLP) to encode and project slice vectors into the vocabulary. These logits are then added to the LM's before taking the softmax to obtain the final next-token prediction distribution. During training, tokens are sampled from a categorical distribution and contribute to the VAE's overall ELBO loss. While this technically is a slight difference to P+22, who used categorical cross-entropy loss, we are able to closely reproduce their reported baseline perplexity on the test set ($\approx 46 \pm 0.1$). As the language modeling baseline we finetune GPT-2 in the target domain (on the raw WSJ text) without any exposure to SGS, as did P+22.

4.3 Evaluation

We measure language modeling performance in terms of surprisal or *perplexity* (PPL), which is computed as the exponential of the model's token-averaged negative log-likelihood (NLL).[6] Whenever we report aggregate performance over all data, we use PPL (tables 2 and 3), but in the detailed analysis of smaller subsets of data we switch to NLL for better readability (fig. 6). For both metrics, lower is better. To evaluate B-SGS correctness, we consider binary micro-accuracy over individual vector dimensions, macro-accuracy over tokens, as well as edge precision, recall, and F1-score.

5 Representation Distillation: What makes semantic structure valuable to language modeling?

Currently well-known as a popular and effective deep learning technique (e.g., Polino et al., 2018; Sanh et al., 2019), distillation (of neural models)

---

[5]https://github.com/jakpra/LinguisticStructureLM
[6]See Limitations section for shortcomings of this metric.

aims to reduce redundancy and unwieldiness (§5.1) while retaining core information. Here we apply a similar concept to a family of *symbolic linguistic* representations, SGS. Rather than relying on a blackbox training process to transfer knowledge from a large pretrained model to a smaller model, we manually design a less detailed variant of SGS, which we call B-SGS (§5.2). We use ground-truth B-SGS as additional input to the incremental LM as before and find that it *does* constitute a **lower bound of sufficient richness** (§5.3).

5.1 Unparsimoneousness of Fully Labeled and Anchored SGS

While the very rich SGS representation used by P+22 (which, here, we call F(ull)-SGS; §2.1 and fig. 2 top) proved to be a very potent next token predictor, this power comes at the cost of being rather unwieldy and, as it turns out, redundant.

As input. Recall from §2.1 that, in F-SGS, preceding tokens that are semantically related to each target token (via edges in the graph) are encoded by concatenating their embeddings (in a specific order and with zero-padding to preserve their structural relation, e.g., parent vs. sibling, see §2.1). It is obvious at first glance that this quickly leads to very large slices and models (P+22 report an average influx in models size of 50-60 million parameters for SGS encoding alone). Furthermore, linguistic formalisms vary greatly in the number of semantic relation types (edge labels) they distinguish: e.g., 10 in EDS vs. 72 in PTG. And while this number does not seem to be directly associated with model performance, it still makes the comparison somewhat blurry. In addition to their excessive size, F-SGS vectors also seem to be partially redundant with a pretrained LM, since P+22 found in their ablation experiments that the correct edge label assignment is not essential for achieving high language modeling performance.

As output. In addition to oracle-augmented language modeling, a major use case of SGS we work towards is to incrementally predict them (cf. §1 and §6). This is, however, a non-trivial structured prediction problem. It consists at least of edge prediction and relation classification (cf. Liu et al., 2019). And while on the surface, this is reminiscent of a task that could be solved with an edge-factored parser (Kiperwasser and Goldberg, 2016; Dozat and Manning, 2017), our scenario is much more complex due to the multitude of structural relations
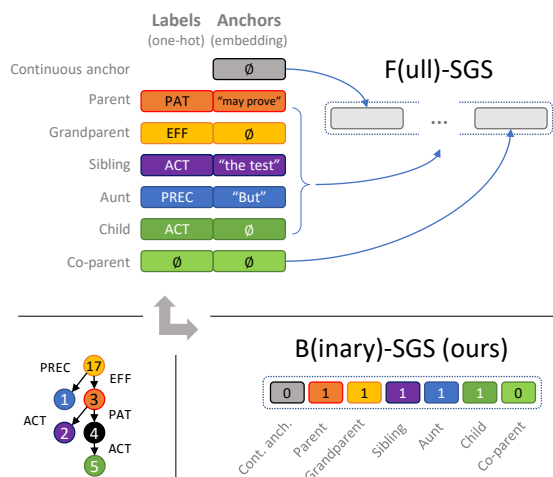

Figure 2: Deriving **F**ull and **B**inary semantic graph slice (SGS) vectors from the PTG subgraph for the token 'to' in fig. 1. The *continuous anchor* dimension would be filled, e.g., in the SGS for the tokens 'test' and 'prove', which each share the rest of their slice with their respective preceding tokens. Node ⑤ in the slice for 'may prove' is an example of a co-parent.

(not just parents), the possibility of multiple parents for each node, abstract nodes not directly anchored in a single token, as well as incrementality. Indeed, it is more akin to supertagging (Bangalore and Joshi, 1999; Clark and Curran, 2004, §2.2) but without the formal guarantees of a mildly context-sensitive grammar formalism like TAG or CCG. In our early exploration with simple multilayer perceptron (MLP) classifiers and a combination of loss functions (categorical cross-entropy for labels; cosine similarity and/or attention loss for token-to-token anchoring), we found it very difficult to train a model to convergence. We suspect that full SGS prediction warrants more complex modeling, optimization, and inference mechanisms, which we leave to future work.

5.2 Reducing SGS to Binary Structural Relations

The challenges described above prompt us to drastically simplify the SGS encoding. We propose to collapse both edge labels and anchor-token embeddings into mere binary indicators of whether an edge of a given *structural* relation type (flat subword continuation, parent, sibling, grandparent, aunt, child, co-parent) exists, resulting in *binary* semantic graph slices, or B-SGS (fig. 2).[7]

---

[7]While each node may have multiple relatives of the same type (e.g., 2 parents and 3 siblings), a single binary dimension for each type could only indicate the existence of *at least* one

	A	P	P+	O	S	T	C	R
UD	.11	.34	0.	.24	.31	.17	.34	0.
DM	.33	.48	.21	.19	.24	.09	.25	.21
PTG	.43	.75	.07	.69	.42	.41	.41	.10
EDS	.30	.69	.51	.17	.29	.09	.26	.22

Table 1: Relation-wise density of B-SGS vectors in the development set. A: continued anchor, P: parent, P+: 2 or more parents, O: grandparent, S: sibling, T: aunt, C: child, R: co-parent. UD and DM are shown for reference (cf. Liu et al., 2019, §6.1).

Data statistics. We report average density of major SGS dimensions (= relation types) in table 1. Note in particular that EDS and PTG differ substantially in the types of structures they encode, with PTG being denser on average. EDS is quite similar to DM because they are both derived from the same underlying formalism. In contrast to EDS, PTG, and DM graphs, which are generic DAGs, UD graphs are strictly bilexical dependency trees, leading to necessarily empty P+ and R dimensions.

5.3 Validating LM Performance with Oracle B-SGS

Setup. We train for up to 10 epochs, with early stopping based on development set perplexity. See §4 for more details.

Results. Table 2 shows that although B-SGS perplexity is slightly worse than with F-SGS—which is to be expected given the drastic reduction of the input signal—it still clearly outperforms the non-symbolic baseline. This suggests that the most crucial signal contributed by SGS in general is, in fact, the bare-bones hierarchical structure itself. And while P+22's ablation analysis already suggested that the grouping into different edge labels may be less important, it is quite surprising that even the information about *which other tokens* the target token is hierarchically-related to is not necessary to improve language modeling with SGS.

A possible explanation can be found in the fact that the baseline LM already has extensive *gradient* representation of parts-of-speech, syntactic functions, and semantic roles (namely, in its dense hidden states and attention distributions). What it

such instance. We follow P+22 in allocating additional 'low-resolution' dimensions for certain relation types to indicate the existence of 2 or more relatives. This is illustrated for parents (P+) in table 1 but otherwise omitted (e.g., from fig. 2) for simplicity. Note that a node having multiple parents is distinct and independent from it having one or more *co-parents* (i.e., other parents of the node's children).

	PTG	EDS
Pretrained GPT-2	59.3	
Domain-finetuned (baseline)	46.1	
Gold F-SGS	26.8	24.7
Gold B-SGS (ours)	33.9	28.0

Table 2: Comparing test set LM perplexity (lower is better) with our **B**inary slices against **F**ully labeled/ anchored ones (P+22).

might be lacking, then, is any *discrete* representation, and in particular a commitment to discrete and complex semantic structure seems to be beneficial.

Gold B-SGS is thus a sufficiency lower bound.

6 Noise Robustness: How accurate should bootstrapped semantic structure be in order to improve a LM?

In a pilot experiment, we integrate into the P+22 model B-SGS *prediction*. As illustrated in fig. 3, this is an intermediate step, the output of which is now used as input to next-token prediction instead of the ground truth slice. We find that while our relatively simple model (§6.1) produces B-SGS outputs of seemingly reasonable overall quality (in terms of micro-accuracy and F-score), they are *not* sufficient for supporting LM performance. This prompts us to actively search for **lower bounds of sufficient correctness** by artificially inducing various types and levels of noise into gold B-SGS inputs (§6.2). We do find several bounds, but learn that what makes them sufficient has less to do with their single-number correctness and more with intricate details of their overall noise distribution (§6.3).

6.1 Pilot Prediction

Setup. Since we are interested in lower bounds and we are running an exploratory study, we do not perform extensive model engineering. The following description is purely intended for clarity and replicability rather than as a state-of-the-art model proposal. We decide on a variational autoencoder (VAE; Kingma and Welling, 2013), where sampling from the latent space mediates between the LM's hidden state and the sigmoid-activated B-SGS dimensions (fig. 3). This setup is motivated by the high uncertainty involved in the task (we predict the symbolic structure of a token that has not been observed yet, and there may be much genuine ambiguity). All encoders, decoders, and projectors within the VAE, besides GPT-2, are simple
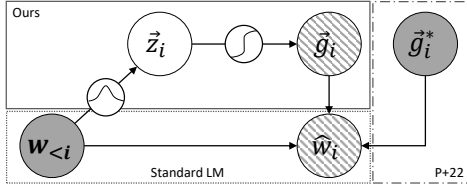
Figure 3: Our simple variational autoencoder model. We project the encoding of observed (solidly shaded) previous words $\mathbf{w}_{<i}$ into latent space and sample hidden states $\vec{z}_i$. Predicted graph slices $\vec{g}_i$ and target tokens $\hat{w}_i$ are supervised during training but unobserved at test time. P+22 used ground truth slices $\vec{g}_i^*$ instead of predicted ones. The standard LM is a component in both versions.

feed-forward MLPs. B-SGS prediction is trained deterministically with binary cross-entropy loss.[8] We train the slice predictor for up to 10 epochs with early stopping based on dev set F-score, and then train the SGS-augmented LM as before.

Results. As shown in fig. 4, SGS prediction performance is best in layers 8 (PTG) and 9 (EDS). This is in line with previous studies on probing semantic structure (e.g., Liu et al., 2019; Jawahar et al., 2019; Tenney et al., 2019a), which obtained the best performances in middle/high layers. However, even these best predictions cannot outperform the finetuned LM baseline in the augmented language modeling setting (compare black solid and red dashed lines in fig. 5).

Validation. Prediction micro-accuracies (.84 for PTG, .90 for EDS; last row table 3) are in the same order of magnitude as Liu et al. (2019)'s binary edge prediction results for UD and DM, two representation frameworks featured in the literature much more frequently than PTG and EDS. Although there are many differences in task and experimental setup (dependencies vs. constituencies, single-parent vs. B-SGS prediction; cf. table 1), we find this to be a valuable sanity check for both us and the reader in lieu of a proper baseline.

6.2 Artificial Noise

Why is our pilot system not sufficient? Maybe prediction accuracy just needs to be better? We investigate by using shuffled gold B-SGS as inputs to the LM and systematically altering several characteristics of the shuffling routine. This style of

---

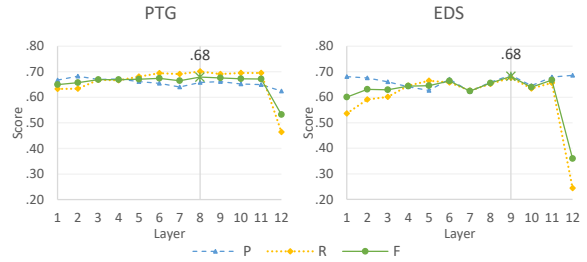[8]We also experimented with Bernoulli sampling, but to no success.

Figure 4: Graph slice prediction performance on the development set by LM layer. P = precision (proportion predicted edges correct), R = recall (gold edges predicted), F = F1-score (harmonic mean of P and R).

control task is inspired by Hewitt and Liang (2019); Dubossarsky et al. (2018).

We consider two different shuffling mechanisms: (a) Shuffling the node-to-word **anchor** mapping of graphs before vector extraction (i.e., which slice corresponds to which word token, cf. P+22). This guarantees well-formed graphs but may be too optimistic since we only shuffle within each sentence. Thus we also consider a more aggressive option: (b) Randomly switching **bits** (= whether or not a given edge type exists) in the slice vectors extracted from gold graphs.

For both, we also produce *varying degrees* of noise. Namely, whenever we are about to shuffle a graph anchor or vector bit, we decide to instead retain the correct assignment with probability p_{Gold}.

Results. Table 3 shows how the different shuffling conditions affect B-SGS correctness. As expected, within-sentence graph anchor shuffling is generally much more optimistic than bit-switching. By definition, p_{Gold} directly determines micro-accuracy in bit-switched slices, whereas in anchor-shuffled slices, p_{Gold} is more closely correlated with macro-accuracy. LM perplexity of each condition is shown in fig. 5. Note that the signal strength of bit-switching is symmetric around .5. This is an intuitive corollary of it being a binary signal (though macro-accuracy and F-score naturally continue to decline with $p_{Gold} < .5$, as shown exemplarily for values .1 and 0.).

First, we identify conditions that beat the domain-finetuned LM baseline from fig. 5, and then consult table 3 to find their corresponding slice quality. This results in the following sufficiency lower bounds (marked with asterisks in table 3): **Shuffled graphs** with $p_{Gold} \in \{.9, .8, .7, .6, .5\}$ for both PTG and EDS as well as $p_{Gold} \in \{.4, .3, .2, .1, 0.\}$ for EDS; and **bit-switched vec-**
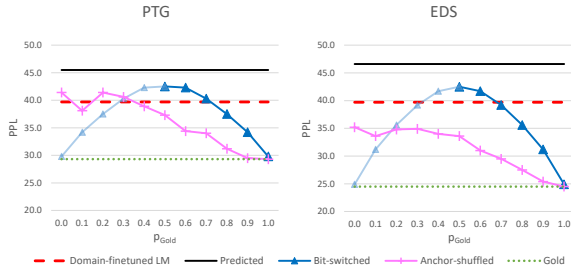
Figure 5: Dev set perplexity (lower is better) of noise-interpolated B-SGS LMs.

	PTG				EDS			
	MaA	MiA	F	LB	MaA	MiA	F	LB
Gold	1.00	1.00	1.00	*	1.00	1.00	1.00	*
Shuffled graph anchors								
p_{Gold} = .9	>.99	>.99	>.99	*	.95	.99	.97	*
p_{Gold} = .8	.88	.97	.94	*	.81	.97	.91	*
p_{Gold} = .7	.60	.93	.86	*	.65	.95	.84	*
p_{Gold} = .6	.54	.92	.83	*	.53	.93	.76	*
p_{Gold} = .5	.36	.87	.72	*	.34	.88	.63	*
p_{Gold} = .4	.23	.83	.64	?	.25	.87	.56	*
p_{Gold} = .3	.18	.81	.60	−	.19	.85	.49	*
p_{Gold} = .2	.12	.80	.56	−	.17	.84	.48	*
p_{Gold} = .1	.10	.79	.55	?	.14	.83	.44	*
p_{Gold} = 0.	.08	.78	.53	−	.13	.82	.41	*
Bit-switched vectors								
p_{Gold} = .9	.17	.90	.81	*	.17	.90	.75	*
p_{Gold} = .8	.02	.80	.66	*	.02	.80	.58	*
p_{Gold} = .7	<.01	.70	.53	?	<.01	.70	.44	?
p_{Gold} = .6	<.01	.60	.42	−	<.01	.60	.34	−
p_{Gold} = .5	0.	.50	.32	−	0.	.50	.26	−
⋮			⋮				⋮	
p_{Gold} = .1	0.	.10	.05	*	0.	.11	.05	*
p_{Gold} = 0.	0.	<.01	<.01	*	0.	.01	.02	*
Predicted	.18	.84	.68	−	.29	.90	.68	−

Table 3: Correctness F1-score (F), accuracy at the macro (token-level, MaA) and micro (bit-level, MiA) levels of B-SGS with various levels of noise (measured on the dev set). The LB columns indicate whether a condition is a sufficiency lower bound (*=yes, ?=maybe, −=no), i.e., if its corresponding PPL beats the baseline (fig. 5).

tors with $p_{Gold} \in \{.9, .8, .2, .1, 0.\}$, which can be generalized $|p_{Gold} - .5| \geq .3$.

6.3 Detailed Analysis

Unexpectedly, shuffled slices with clearly worse overall accuracy than predicted ones (table 3) sill yield much better perplexity (fig. 5). This leads us to the following hypotheses which we address in order. For brevity, we focus only the comparison between *predicted* and *bit-switched with $p_{Gold} = .8$*, because this condition seems to be a good trade-off between matching or slightly beating the PPL baseline and realism in terms of closeness to predicted in terms of overall accuracy. Consider fig. 6.

Hypothesis: The noise of shuffled slices is uniformly distributed over tokens whereas the noise of predicted slices is distributed similarly as baseline LM surprisal.
Average F-score of predicted B-SGS does indeed decrease as baseline LM surprisal increases (fig. 6b). However, contrary to our expectation, the *same* is true for the F-score of uniformly shuffled slices (fig. 6d)! Thus, the distribution of F-score means over suprisal bins alone does not explain the difference.

Hypothesis: Due to high-surprisal tokens having low B-SGS correctness, we create a noisy feedback loop which worsens LM surprisal in particular for already high-surprisal words (open-class content words) without gaining enough advantage on low-surprisal words.
We find quite the opposite: Both predicted and shuffled slices help in particular for very-high-surprisal tokens, despite the higher average slice noise (fig. 6c+e). In contrast, predicted slices tend to slightly increase surprisal for low-surprisal tokens. And since low-BL-surprisal tokens make up the vast majority of the data (fig. 6a), this slight increase might be enough to confuse the LM beyond baseline. Another crucial factor might be *variance*

in slice correctness, which is generally much higher in predicted slices than in shuffled ones (fig. 6b+d).

Most affected words. We manually inspect the data to get an idea of how predicted B-SGS benefits the LM the most. The top 10 tokens in terms of both baseline NLL and Δ NLL (bottom right region of fig. 6c) are dominated by (recurring) named entities and dates, which are likely just an artifact of overfitting. After filtering these out, we find that the highest-baseline-NLL tokens are mostly nouns, adjectives, and verbs that are either rare themselves (e.g., *hopscotched*, *instrumentation*) or used in a rare construction (*paying thin **compliments***). In contrast, both PTG and EDS B-SGS reduce NLL the most for verbs, particularly in participle constructions (*dividing*, has **begged**, will be **relocated**).

7 Discussion and Conclusions

We proposed a general framework for semantically-enriched language modeling. Our proposal aims to provide a new perspective on qualitative distributional linguistic analysis, expanding upon prior work in linguistic analysis of neural models in sev-
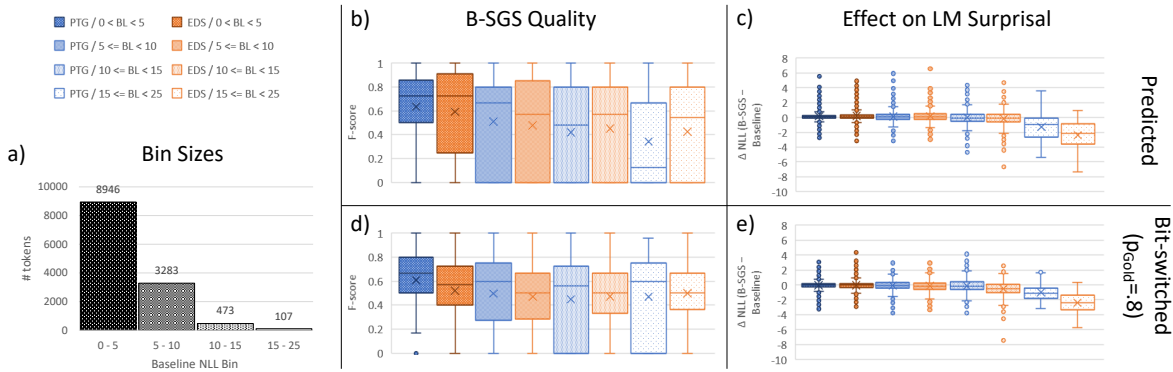
Figure 6: Box-and-whiskers plots of distributions over dev set tokens of B-SGS quality (F1-score) and effect on LM surprisal (Δ NLL), binned by baseline (BL) NLL. $\times$ markers are means and lines within boxes are medians.

eral ways (§7.1). We implemented and tested this framework with GPT-2 and semantic graph slices (SGS) from two formalisms, finding interesting patterns with potential impact for meaning representation design and low-resource modeling (§7.2).

7.1 General Framework

Probing and related methods evaluate language models based on their ability to predict linguistic representations from text. Although this is a relatively practical setup which has already produced many fascinating and replicable findings, it has the disadvantage that results need to be interpreted relative to both the linguistic framework governing the output and the specific probing architecture. In contrast, the approach of Prange et al. (2022) takes linguistic representations as an input and evaluates the language model directly on its native language modeling task. The main problem with their oracle setup is that it is unrealistic to have ground truth linguistic structures available at test time.

We argue for unifying the advantages of both directions, by considering what is in essence a concatenation of the two: a pipeline in which the output of a structure prediction model (similar to probing, except that it may be supervised) is fed back into the LM, enabling comparable evaluation on the raw text itself. This makes it possible to identify shortcomings of the LM and/or benefits of the linguistic representation quantitatively and qualitatively, by modifying either the probing architecture or the linguistic representation itself until LM performance starts or stops improving. The *lower bounds* of this continuum in particular (in contrast to upper bounds) have many theoretical and practical implications, since they separate the wheat from the chaff when it comes to the effi-

ciency/effectiveness trade-off for model and representation. Our definition of sufficiency lower bounds in terms of the signal's data distribution in §3 is intentionally kept high-level and flexible to stimulate adaptations of the idea for a variety of use cases. While here we take an exclusively empirical approach, the framework may lend itself to formally-provable accounts as well.

7.2 Concrete Take-aways

In our experiments with GPT-2 (§5 and §6), we were able to crystallize the simple (unlabeled and unanchored) discrete hierarchical semantic structure of PTG and EDS as both beneficial to language modeling and robust to certain types of noise. We also found, though, that measuring prediction quality via a single aggregate score hides important aspects of the distributions of signal and noise, to the extent of potentially nullifying LM improvements. While the respective structures of PTG and EDS differ from each other in terms of density, relations encoded (§5.2), prediction accuracy (§6.1), and LM benefit (§6.2), the types of words they help the LM with the most are similar (§6.3).

As a nice side-effect from §5, removing the explicit token anchoring from SGS also makes it more applicable to unanchored semantic representations such as AMR (Banarescu et al., 2013). Note, however, that we still need some source of basic anchoring information (e.g., from an automatic aligner) in order to assign a slice to each token.

Finally, based on our findings in §6.3 that rare high-surprisal words most positively affected by even noisily SGS-enhanced language modeling, we are hopeful that our method may be particularly helpful for the Zipfian tail at a small cost to the majority of data.

464

Limitations

No guarantees. As stated in §3 and substantiated in §6, sufficiency lower bounds tend to be nontrivial, multifaceted configurations. We explore this to some extent (we find, e.g., that overall correctness scores alone, without variance, are *not* reliable identifiers of sufficiency lower bounds), but not exhaustively. To make stronger guarantees rather than just optimism, we need to precisely define when a candidate distribution is 'similar enough' to a known lower bound (e.g., via goodness-of-fit).

Practicability of semantic bootstrapping. We do not present a complete working system yet. It could be that sufficiently distributed performance can only be achieved with more intricate structured decoding mechanisms (e.g., Viterbi or beam search), which would negatively affect running time and thus usability as an end-to-end LM.

Limited evaluation of LM quality. Our evaluation of LM quality has been limited to the effects of the predicted graph slices on the perplexity metric. Alternative evaluations adopting psycholinguistically-inspired metrics, such as the correlation with human norms collected from cloze completion tasks, might yield different results (Hao et al., 2020).

Acknowledgements

References

Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. 2021. Can language models encode perceptual structure without grounding? a case study in color. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 109–132, Online. Association for Computational Linguistics.

Bharat Ram Ambati, Tejaswini Deoskar, Mark Johnson, and Mark Steedman. 2015. An incremental algorithm for transition-based CCG parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 53–63, Denver, Colorado. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. 2018. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*.

Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The Prague Dependency Treebank: A three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, Text, Speech and Language Technology, pages 103–127. Springer Netherlands, Dordrecht.

Gemma Boleda and Aurélie Herbelot. 2016. Formal distributional semantics: Introduction to the special issue. *Computational Linguistics*, 42(4):619–635.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Emmanuele Chersoni, Enrico Santus, Chu-Ren Huang, and Alessandro Lenci. 2021. Decoding Word Embeddings with Brain-based Semantic Features. *Computational Linguistics*, 47(3):663–698.

Stephen Clark and James R. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 282–288, Geneva, Switzerland. COLING.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2):281–332.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Haim Dubossarsky, Eitan Grossman, and Daphna Weinshall. 2018. Coming to your senses: on controls and evaluation sets in polysemy research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1732–1740, Brussels, Belgium. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

Adam Ek, Jean-Philippe Bernardy, and Shalom Lappin. 2019. Language modeling with syntactic and semantic representation for sentence acceptability predictions. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 76–85, Turku, Finland. Linköping University Electronic Press.

Guy Emerson. 2020. What are the goals of distributional semantics? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7436–7453, Online. Association for Computational Linguistics.

John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Dan Flickinger. 2000. On building a more effcient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Proc. of NeurIPS*.

Goran Glavaš and Ivan Vulić. 2021. Is supervised syntactic parsing beneficial for language understanding tasks? an empirical investigation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3090–3104, Online. Association for Computational Linguistics.

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3153–3160, Istanbul, Turkey. European Language Resources Association (ELRA).

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2727–2736, Melbourne, Australia. Association for Computational Linguistics.

Yiding Hao, Simon Mendelsohn, Rachel Sterneck, Randi Martinez, and Robert Frank. 2020. Probabilistic Predictions of People Perusing: Evaluating Metrics of Language Model Performance for Psycholinguistic Modeling. In *Proceedings of the EMNLP Workshop on Cognitive Modeling and Computational Linguistics*.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Hany Hassan, Khalil Sima'an, and Andy Way. 2009. Lexicalized semi-incremental dependency parsing. In *Proceedings of the International Conference RANLP-2009*, pages 128–134, Borovets, Bulgaria. Association for Computational Linguistics.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

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

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive explanations for model interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1597–1611, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Najoung Kim, Roma Patel, Adam Poliak, Patrick Xia, Alex Wang, Tom McCoy, Ian Tenney, Alexis Ross, Tal Linzen, Benjamin Van Durme, Samuel R. Bowman, and Ellie Pavlick. 2019. Probing what different NLP tasks teach machines about function word

comprehension. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 235–249, Minneapolis, Minnesota. Association for Computational Linguistics.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Nikita Kitaev, Thomas Lu, and Dan Klein. 2022. Learned incremental representations for parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3086–3095, Dublin, Ireland. Association for Computational Linguistics.

Alessandro Lenci. 2023. Understanding natural language understanding systems. a critical analysis. Preprint arXiv:2303.04229.

Tao Li and Vivek Srikumar. 2019. Augmenting neural networks with first-order logic. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 292–302, Florence, Italy. Association for Computational Linguistics.

Xiang Lisa Li and Alexander Rush. 2020. Posterior control of blackbox generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2731–2743, Online. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

William Merrill, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. 2021. Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? *Transactions of the Association for Computational Linguistics*, 9:1047–1060.

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, and Daniel Zeman. 2020. MRP 2020: The second shared task on cross-framework and cross-lingual meaning representation parsing. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1–22, Online. Association for Computational Linguistics.

Stephan Oepen, Omri Abend, Jan Hajic, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdenka Uresova. 2019. MRP 2019: Cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1–27, Hong Kong. Association for Computational Linguistics.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Juri Opitz and Anette Frank. 2022. SBERT studies meaning representations: Decomposing sentence embeddings into explainable semantic features. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 625–638, Online only. Association for Computational Linguistics.

Antonio Polino, Razvan Pascanu, and Dan Alistarh. 2018. Model compression via distillation and quantization. In *International Conference on Learning Representations*.

Jakob Prange, Nathan Schneider, and Lingpeng Kong. 2022. Linguistic frameworks go toe-to-toe at neuro-symbolic language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4375–4391, Seattle, United States. Association for Computational Linguistics.

Jakob Prange, Nathan Schneider, and Vivek Srikumar. 2021. Supertagging the long tail with tree-structured decoding of complex categories. *Transactions of the Association for Computational Linguistics*, 9:243–260.

Peng Qian, Tahira Naseem, Roger Levy, and Ramón Fernandez Astudillo. 2021. Structural guidance for transformer language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3735–3745, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. Preprint arXiv:1910.01108.

Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. 2022. Transformer Grammars: Augmenting Transformer Language Models with Syntactic Inductive Biases at Scale. *Transactions of the Association for Computational Linguistics*, 10:1423–1439.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pages 593–607, Cham. Springer International Publishing.

Petr Sgall, Eva Hajicová, and Jarmila Panevová. 1986. The meaning of the sentence and its semantic and pragmatic aspects. academia.

Miloš Stanojević, Shohini Bhattasali, Donald Dunagan, Luca Campanelli, Mark Steedman, Jonathan Brennan, and John Hale. 2021. Modeling incremental language comprehension in the brain with Combinatory Categorial Grammar. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 23–38, Online. Association for Computational Linguistics.

Miloš Stanojević and Mark Steedman. 2019. CCG parsing algorithm with incremental tree rotation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 228–239, Minneapolis, Minnesota. Association for Computational Linguistics.

Miloš Stanojević and Mark Steedman. 2020. Max-margin incremental CCG parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4111–4122, Online. Association for Computational Linguistics.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective inference for generative neural parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700, Copenhagen, Denmark. Association for Computational Linguistics.

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

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *Proc. of ICLR*.

Sean Trott, Tiago Timponi Torrent, Nancy Chang, and Nathan Schneider. 2020. (Re)construing meaning in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5170–5184, Online. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Pia Weißenhorn, Lucia Donatelli, and Alexander Koller. 2022. Compositional generalization with a broad-coverage semantic parser. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 44–54, Seattle, Washington. Association for Computational Linguistics.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online. Association for Computational Linguistics.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 184–198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.