

1. Introduction

Large language models, or LLMs, have a huge capacity for producing and processing natural language. Given a prompt, modern models can output a stream of words that read as cohesive and natural as human output. They have learned to produce language often indistinguishable from ours, a feat once considered a piece of intelligent behavior (Turing, 1950). However, we cannot be sure if they have developed this ability with human-analogous methods. Knowing this is necessary to interpret these opaque models and see how they can actually contribute to our scientific understanding of human language and language processing.

For example, we can wonder whether the LLM syntactically parses the sentence or represents syntax at some level of its representation. Tian et al. (2024) investigate the ability of LLMs to succeed at constituency parsing, or syntactically breaking down a sentence into chunks. Models are only trained to create embeddings for individual words (or tokens), and not for larger phrases. LLMs are normally ineffective at these parsing tasks, producing very shallow parse trees and preferring shorter chunks over longer ones. However, Tian et al. show that using careful chain-of-thought prompting, models can learn to create valid parse trees and correctly identify constituents. Results such as these suggest LLMs can be “taught” to show their syntactic knowledge, although it is still unclear if the syntax exists in the model without precise prompting.

Hewitt & Manning (2019) present a different methodology for finding out whether an LLM represents syntactic structure. Rather than looking at the output, or if its next-word prediction is correct, they actually look at the representation an LLM has made of the sentence. By sending a structural probe into the representation, they test if there is some layer that looks like it is representing syntax, specifically human-annotated syntactic dependencies between the tokens of the sentence. After generating such representations from the model and comparing them to ground truth syntactic parses, they report that BERT (Devlin et al., 2019) is actually encoding syntax “through our structural probes”.

This is a hugely impactful result, as it would imply that neural networks do in fact abstract syntactic information from language in a human-like way. Neural networks are often extremely obscure, with billions of weights and hundreds of dimensions per word. In these spaces, we can only talk about relative directions and theorize about small slices of the network. There is no way a human can comprehend everything a neural network does. Therefore, showing that it represents syntax would contribute to arguments that LLMs are capable of abstracting in human-like ways (Kim et al., 2023; Wang et al., 2025)¹.

One issue lurking in the shadows of the Hewitt & Manning method is semantics. Another way that the dependency between, say, “green” and “tree” could emerge via the probe is if the model

¹ In the six years since its publication, Hewitt & Manning has been cited 1,360 times per Google Scholar, frequently as evidence for LLMs learning or representing syntax.

is simply representing them as related semantically, rather than syntactically as an adjective modifying a noun. If the layer was truly representing syntax, we should see the same trees emerge for the same syntactic structures no matter the semantic relationships between the words; “green” and “tree” should have the same relationship as “consular” and “tree” would. Therefore, Hewitt & Manning’s methodology is missing an absolutely critical piece. Only if the structural probe works for sentences that, despite being semantically anomalous, are still syntactically correct, can we confirm that it is finding a syntactic tree.

In this paper, we propose a methodology to fill this piece and clarify the meaning of this tree in the network’s representation space. We generate semantically anomalous sentences from a corpus and test the exact code Hewitt & Manning ran on them. We find that there is reason to think that Hewitt & Manning’s probes are using some semantic information, as the probes regularly fail to find the same dependencies on the anomalous sentences.

2. Hewitt & Manning’s Structural Probes

Deep learning has progressed from less to more use of context in capturing words’ roles in sentences. Word2vec (Mikolov et al., 2013) succeeded in using a very small window to develop surprisingly rich lexical meaning spaces. ELMo (Peters et al., 2018) introduced word embeddings that are a function of the entire input sentence: a crucial aspect of this training is that it uses a bidirectional LSTM which maximizes the likelihood at a given point of both the forward and the backward directions; e.g. the probability based on both the previous and the next tokens. This allows each word to gain information based on its position in the overall sentence, not just based on the previous tokens.

BERT (Devlin et al., 2019) uses an even more advanced technology to learn the context of each word in the sentence: the transformer. The transformer identifies which information the network should pay attention to at any point in the input. This allows the parts of the input that are important to shine for each specific word and allows the model to capture the context of the input better. Additionally, the model can now focus on distant parts of the input simultaneously, improving its performance on syntactic tests. To create its word representations, BERT was trained with a masked-word prediction task. For a given sentence, some proportion of the words were covered with a mask: “Outside the state capitol, they were raising the MASK”. If the model knew enough about the words to output “flag”, it would succeed at this masked task. This ensured the model had a deep knowledge about each different sense of each word in its vocabulary and its requirements in a syntactic and semantic context.

Contextual appropriateness, however, is not a monolithic concept. Chomsky and decades of continuing research have provided evidence that humans’ syntactic and semantic capacities are distinct: the famous “colorless green ideas sleep furiously” illustrates that a sentence can be identified as syntactically well-formed even if semantically nonsensical, and we can understand “The Force very powerful is” even if it violates rules of English syntax.

To investigate the *syntactic* abilities of these newer models, it is therefore not enough to just look at the next-word prediction or the output, as they are extremely capable at producing natural output. Instead, Hewitt & Manning (2019) decided to look at their new innovation: their vast

word representations. The model was not given trees as input, just huge corpora, so this is not finding anything that is encoded explicitly in the input; any trees that are found must be created in the model's representation, using its implicitly encoded knowledge of language. Hewitt & Manning created a transformation to find the slice of the network that best encoded a tree, arguing that if these tree structures matched human parses, this provided evidence for LLMs encoding human syntax.

Specifically, they introduced a *structural probe*, which looks for tree structures by finding a transformation between the squared L2 norm of the vectors of two words' representations and their distance in the parse tree. Through supervised gradient descent (shown in the formula below), the distance probe finds the transformation that approximates these properties best. It is important to note that the distance probe is not given a tree as input, nor supervised to reconstruct them, but merely to find a space in which squared distance encodes trees.

$$\min_B \sum_l \frac{1}{|s^l|^2} \sum_{i,j} |d_{T^l}(w_i^l, w_j^l) - d_B(\mathbf{h}_i^l, \mathbf{h}_j^l)|^2$$

The formula above represents their distance probe. $|s^l|$ is the length of the sentence, d_T is the tree distance between the two words, and d_B is the probe's predicted distance based on the words' vector representations \mathbf{h}_i and \mathbf{h}_j . The goal of gradient descent is to optimize the matrix B , the parameters of the probe, to minimize the difference between the probe's predictions and the true parse distance.

$$\min_B \sum_l \frac{1}{|s^l|^2} \sum_i |||w_i|| - ||\mathbf{h}_i||_B^2|$$

To get the depth of a word (the number of edges between that word and the root node), they train a second probe, shown by the equation above. This probe approximates the vector norm of a given word, $||w_i||$, again through gradient descent of the parameter matrix B . Depth is naturally represented by this norm, since both represent an ordering of the words in the sentence. Using both probes' predictions, the tree is recoverable by interpreting any words with distance 1 as neighbors in the result, and the word with the greater depth as the child. Take, for example, the sentence:

A) The chef that went to the stores was out of food.

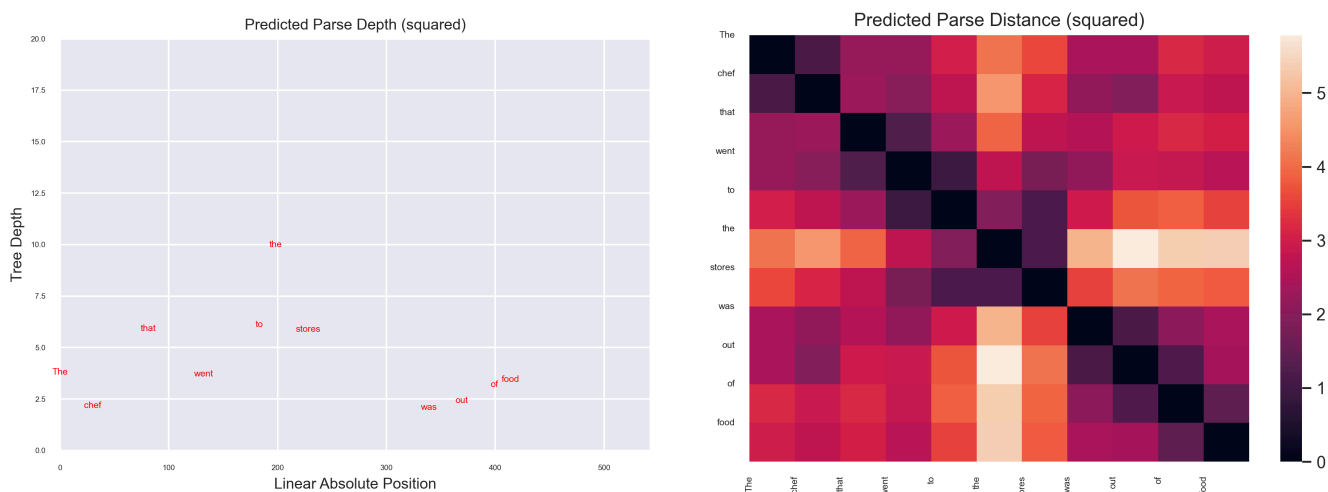


Figure 1: The results of the depth probe (on the left) and the distance probe (on the right) for sentence A. The dependency tree over the sentence can be reconstructed from these results.

The left of Figure 1 shows the depth of each word in sentence A, and the right shows the parse distance between pairs of words. We can see that “the” and “chef” are neighbors, as are “chef” and “out”, using the predicted parse distance. To recreate the predicted tree, we compute the minimum spanning tree of the predicted distances, shown on the bottom of Figure 2.

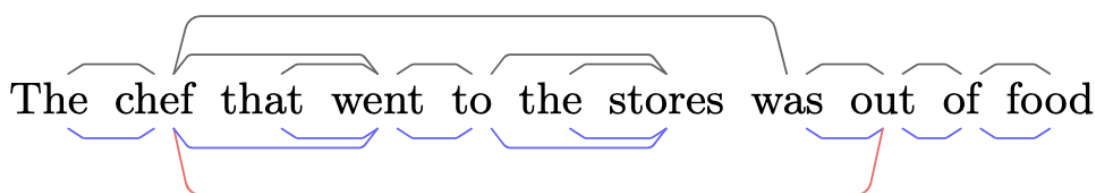


Figure 2: The lines above the sentence display the gold parse, and the lines below display the predicted parse generated by the probe data in Figure 1. Blue lines below agree with the gold parse, while the red line disagrees.

Using the gold parse (from the corpus) and the predicted parse, we can calculate the Undirected Unlabeled Attachment Score (UUAS), the percentage of undirected edges put in the correct place by the probe. In Figure 2, the probe placed 9 of 10 edges correctly, for a UUAS of 90. Overall, their probe obtained an average UUAS of 79.8 on BERT-base and 82.5 on BERT-large-15, as well as a UUAS of 77.0 on ELMo. These results demonstrate the existence of some hierarchical structure encoded in the model’s representation, since neither the probe nor the model were

provided with any structure in their inputs. However, they do not guarantee any results about syntax, since they never demonstrate that their probe actually isolates syntax. Future papers, as well as ours, seek to find how much of this result is boosted by the semantic cues available in the data. This paper is a new contributor to that literature.

3. Work in the Field Since Hewitt & Manning (2019)

Hewitt and Manning’s results have been taken by subsequent literature as strong evidence for syntactic representations in LLMs (Varanasi, Amin, & Neumann, 2020; Lenci & Padó, 2022; Mysiak & Cyranka, 2023). However, their method, and this interpretation, neglect a potentially crucial confound: how can we be certain that the semantics of the words involved does not influence these trees? Deep within the model’s representational space, semantics could be anywhere - every part of this space was given access to the full context of the word during the training, so we cannot escape the idea that the semantics of the word could influence these distance metrics, the context of the sentence, or anything the model may have paid attention to in training. What is captured by the “syntactic” probe may be partly, or even mostly, about semantics, not syntax.

Other papers have investigated the idea that syntax and semantics are intertwined in the results of these probes. One approach to this issue is to transform test sentences to disentangle syntax from semantics. Maudslay & Cotterell (2021) provided the probes not with normal corpus sentences, but with Jabberwocky sentences - named after Carroll’s 1871 poem with the same name, the Jabberwocky corpus contains nonwords with English grammatical markings. These nonwords have not been seen by the LLM before, and contain no semantic information, but still inhabit syntactic roles and can still be placed in a dependency tree.

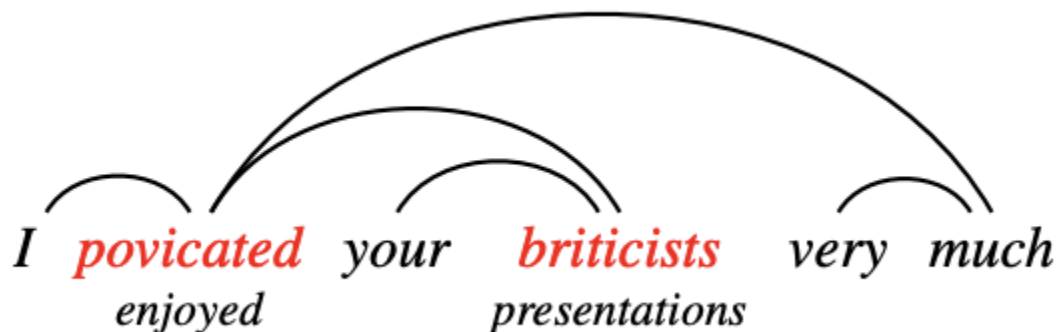


Figure 3: An unlabeled, undirected parse of a Jabberwocky sentence, with the original words below the Jabberwocky words. From Maudslay & Cotterell (2021).

Maudslay & Cotterell used the Universal Dependency (UD; Nivre et al., 2016) treebank to transform sentences using pseudowords from the ARC Nonword Database (Rastle et al., 2002).

They used fine-grained part-of-speech tags to create transformed Jabberwocky sentences that have the same syntactic structure as the original sentence. For example, in Figure 3, “enjoyed” and “povicated” are both VBD (past tense verb), and “briticists” and “presentations” are both NNS (plural noun).

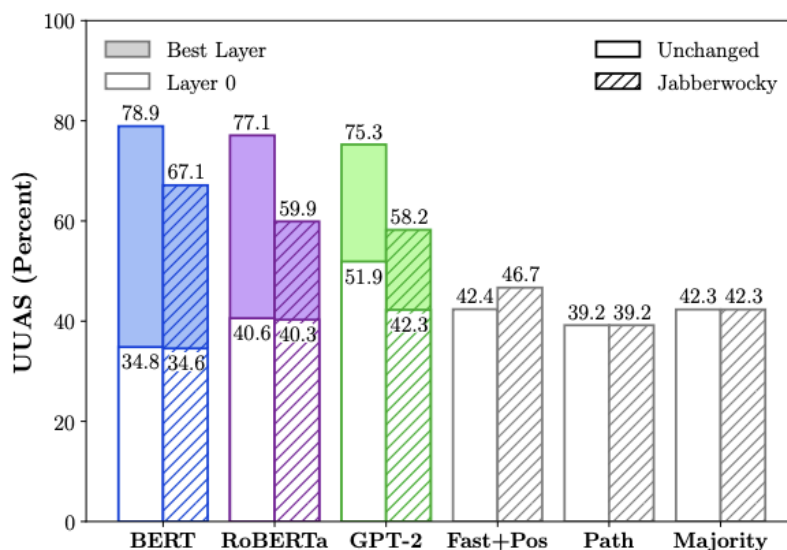


Figure 4: UUAS results from Maudslay & Cotterell’s probing on unchanged sentences vs Jabberwocky sentences. From Maudslay & Cotterell (2021).

If the probe is truly finding syntax, then it should not lose any accuracy when parsing these new, transformed sentences. However, they find a sharp drop in UUAS when testing BERT, from 78.9 to 67.1 (Figure 4). This result strongly suggests that the probes use at least some semantic information during training and output generation. All of the probes they tested performed worse on LLMs, while the baselines (which receive no semantic context or lexical information) performed the same.

This paper still leaves some elements untouched. Since the LLM has not seen any of their replaced words, it can’t have learned any of their specific syntactic or semantic properties. This limits the power of the result, since some of the decrease in probe accuracy may have been because the model had never seen these words before. While the model should be expected to be able to adapt words it knows to new situations, it is not trained to be able to understand non-English words. To clear any doubts about how much impact this had on the result, we need to isolate syntax using only English words.

4. Our Anomalous Idea

To fix these issues with the Jabberwocky study, we propose a new transformation using English words, but in semantically anomalous contexts. We substitute the content words in a sentence with different content words of the same class. For example, a given sentence:

B) The chef baked the tastiest pizza.

has semantic relationships (chef-baked, baked-pizza, tastiest-pizza) in addition to its syntactic relationships. After substitution:

C) The desk considered the sincerest airplane.

these semantic relationships are destroyed, but the syntactic structure remains the same. A human could parse this sentence and imagine sentient furniture having a heartfelt conversation with a vehicle, but this is likely to be nonsense with respect to the model’s semantics. Therefore, its semantic representations should not help it on this task. If we can still see the structural probe having the same effect it did when the semantic relations were present, then we can confirm that the structural probe is isolating syntax. If the tree the structural probe finds does not match the gold parse as well as it did when the semantic relations existed, then this calls Hewitt & Manning’s results into question.

There are several subtleties to consider with this transformation. The main priority is to keep the substituted sentence syntactically correct, with the same exact structure as the original sentence. To control for this as much as possible, we made use of Combinatory Categorical Grammar (CCG), which draws inspiration from lambda calculus to produce an extremely fine-grained syntactic category set (Steedman, 2000; Hockenmaier & Steedman, 2002). Each word in a CCG parse has a syntactic tag and dependencies to other words linking predicates to their arguments. Therefore, a word’s syntactic role is quite well-specified. After confirming that agreement is observed using the syntactic tags, substituting from a bank of words that can be used in the same position and relational situation will generate an appropriate sentence. Figure 5 shows possible substitutions for the sentence “The company is being acquired”, using words that occur in the corpus with the same treebank tag and CCG tag.

						LLR
Penn Treebank Tag		NN			VBN	
CCG Tag		N			S[pss]\\NP	
Original sentence	The	company	is	being	acquired.	6.56
Substitution 1		itinerary			revised.	1.37
Substitution 2		snafu			suggested.	0.79
Substitution 3		freezer			adjourned.	0.02

Figure 5: Possible substitutions alongside their LLR values for the sentence “The company is being acquired”. The best substitution of these is “The freezer is being adjourned”, with an LLR of 0.02.

	A	Not A
B	k11	k12
Not B	k21	k22

Figure 6: A contingency table for words A and B, used in the LLR calculations. More information (and the code used) at <https://github.com/tdunning/python-llr>.

A key aspect of the transformation is that we must ensure the substituted words are semantically distant from the original. We use the Log Likelihood Ratio (LLR) score (Dunning, 1993), which measures the similarity of words based on how often they occur near each other. We measured how often words occurred in the same sentence together. For a given pair of words A and B, we construct a contingency table (Figure 6), counting the number of times they occur together and apart. If these words occur independently, we would expect the first and second rows to be in the same proportion; that is, we would expect the distribution of A given that B is present to be the same as the distribution of A given that B is not present. The LLR of two words is given by

$$LLR(A, B) = 2[D(k11 + k12, k21 + k22) + D(k11 + k21, k12 + k22) - D(k11, k12, k21, k22)]$$

where D, the denormed entropy of a list of numbers C, is given by:

$$D(C) = - \sum_k^C k \log\left(\frac{k}{\sum_i^C i}\right)$$

For sentences that have more than two words to replace, we calculate the maximum LLR between each pair of new words. We then choose the sentence with the lowest of these maxima; the sentence where the most related word pair is the least related among all of the candidate sentences. This will ensure that the substituted sentence is overall composed of semantically anomalous words that did not occur together in the corpus.

6. Methods

To generate our dataset, we make use of the CCGBank dataset (Hockenmaier & Steedman, 2005), which contains Treebank sentences parsed into the CCG format. We use the pipeline in Figure 7. For each sentence, we generate ten candidate sentences with content words substituted with words that have the exact same Treebank and CCG tags. We cleaned up each candidate sentence to fix simple grammar mistakes, such as agreement and capitalization, that could affect how the model parses a given word. We then pick the candidate sentence with the lowest LLR as the transformed sentence. As a confirmation that the sentences are as syntactically identical as possible to the original sentence, each transformed sentence was then put into the spaCY parser (Honnibal & Montani, 2017), and was discarded if its spaCY parse did not match the CCGBank gold parse of the original sentence.

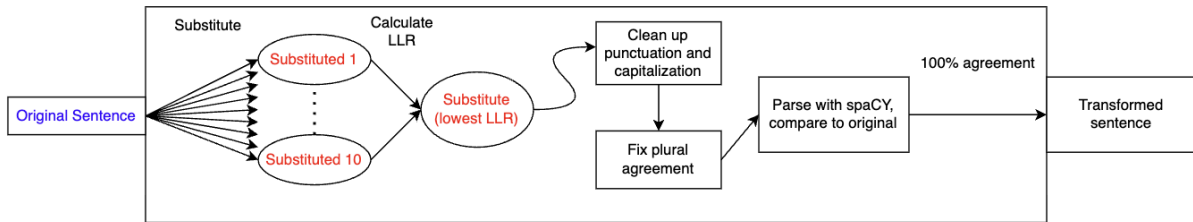


Figure 7: Our pipeline to transform the sentences of CCGBank into semantically anomalous sentences for testing the probe. Code is available at <https://github.com/Innoviox/random-sentences>.

We also used human raters to confirm that our replacement procedure was indeed preserving the syntactic structure. We surveyed two human raters and calculated their inter-rater reliability using Cohen’s kappa (Cohen, 1960). They were presented with the following task for pairs of sentences (original and substituted):

You will have to rate the pairs on a scale from 1 to 5, where 1 means they have completely different structures and 5 means they have identical structures. Try your best to answer each question, but don't spend too long on any one question.

Raters, recruited on the Upwork platform, were required to have prior experience diagramming English sentences, demonstrated by producing a traditional sentence diagram for a sentence they had never seen before. Both raters were native speakers of English and both described themselves as enthusiastic about grammar.

Items were presented in blocks of 20 sentence pairs, plus 5 pairs where the syntactic structure did not match (the replacement was from a different sentence) as attention trials. There were four such surveys for a total of 80 double-rated sentences. Overall, we calculated a kappa value of 0.68, indicating “substantial agreement” (Landis & Koch, 1977), with both reviewers agreeing that 85% of the sentences are syntactic matches (see Appendix for an example survey question). We ran the probe with the original BERT weights using 500 transformed sentences, as well as all 68 of the sentences that human reviewers agreed were syntactic matches. We evaluate the probe using UUAS, the same metric as Hewitt & Manning (2019) and Maudslay & Cotterell (2021)².

7. Results

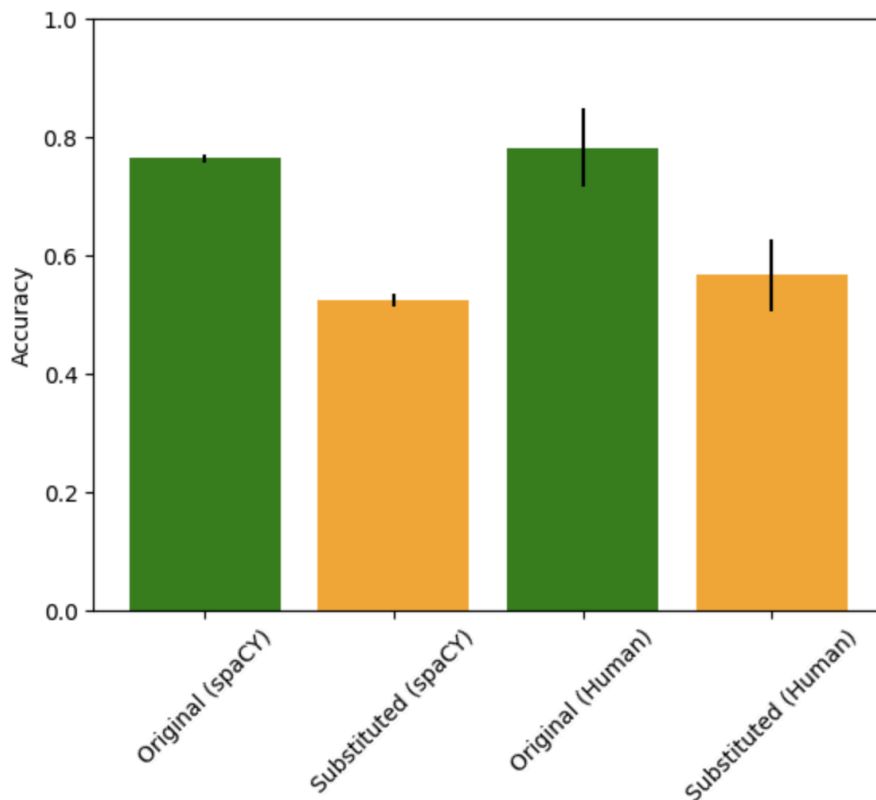


Figure 8: Our results for original sentences (green) and substituted sentences (orange).

² Maudslay and Cotterell also employed a novel probe, the *perceptron probe*. We focus on the original Hewitt & Manning probe since the results of the two probes did not differ substantially in Maudslay & Cotterell’s results.

Figure 8 shows the result of running the probe on our sentences that spaCY confirmed were syntactically identical (the two left bars) and the subset that human reviewers both confirmed were syntactically identical (the two right bars). For the spaCY sentences, we obtained a UAS of 0.764 on the original sentences and 0.524 on the substituted sentences; for the human sentences, we obtained a UAS of 0.782 on the original sentences and 0.566 on the substituted sentences, or a reduction of 31% and 27%, respectively.

For comparison, Hewitt & Manning obtained a UAS of 0.798 on the original sentences, and Maudslay & Cotterell obtained a UAS of 0.789 on the original sentences and 0.671 on the Jabberwocky sentences. The similar values of UAS on the original sentences confirm that we have successfully replicated Hewitt & Manning. The greater reduction in UAS for the semantically anomalous sentences, compared with Maudslay & Cotterell’s Jabberwocky sentences, confirms that Hewitt & Manning’s result can be attributed, to a significant degree, to the model’s knowledge of semantic relationships between human English words.

8. Further Analysis

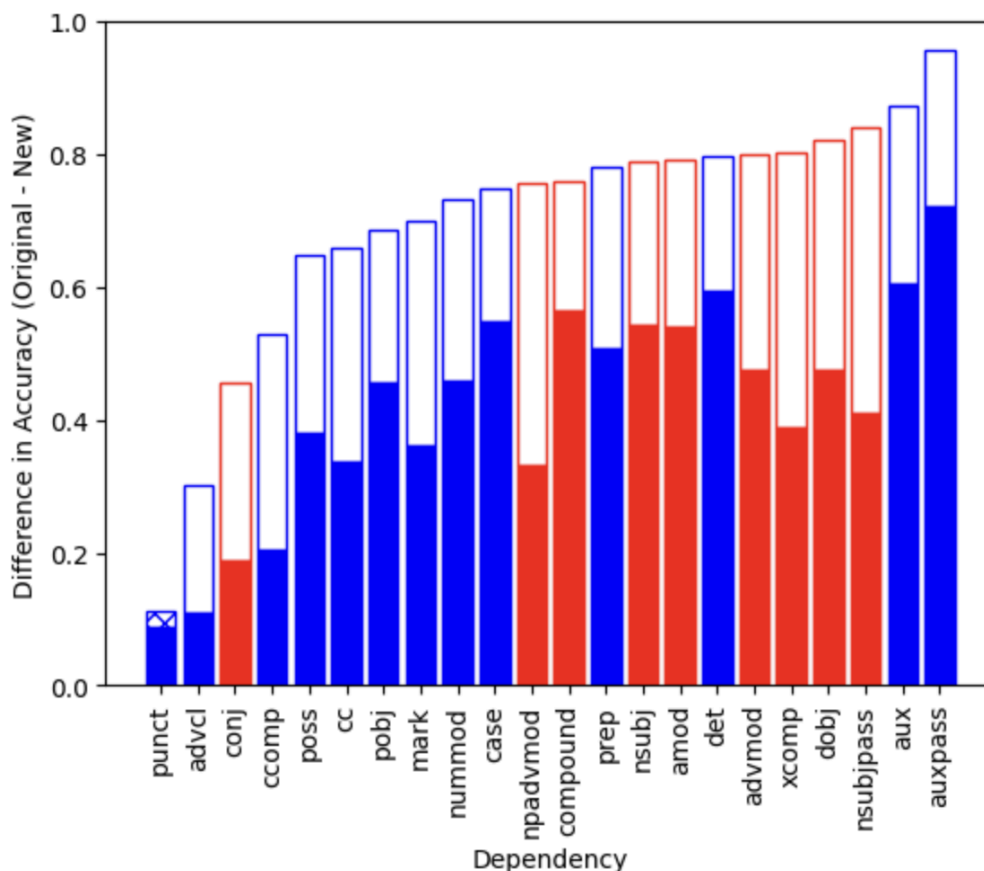


Figure 9: Separating dependencies by what classes of words they connected (blue bars represent dependencies that connect closed-class to open-class words, while red bars represent dependencies that connect open-class to open-class words). White sections of the bar represent loss in accuracy from the original sentence to the substituted sentence.

Parts of speech in English are generally considered to be either closed-class, meaning no new words can be created of that type (e.g. determiners, prepositions), or open-class, meaning new words of the class can be created to encompass new meanings (e.g. nouns, adjectives). For each type of dependency, we calculated how often it connected each pair of classes: closed-open dependencies are blue bars in Figure 9 above, and open-open are red. (Closed-closed dependencies did not show up in significant enough numbers; we only selected dependencies that had at least 100 appearances in the test data.) We would expect open-class words to carry more semantic weight, and therefore for open-open dependencies to degrade more than other types of dependencies under our substitution, since now the semantic link between those types of words should be gone.

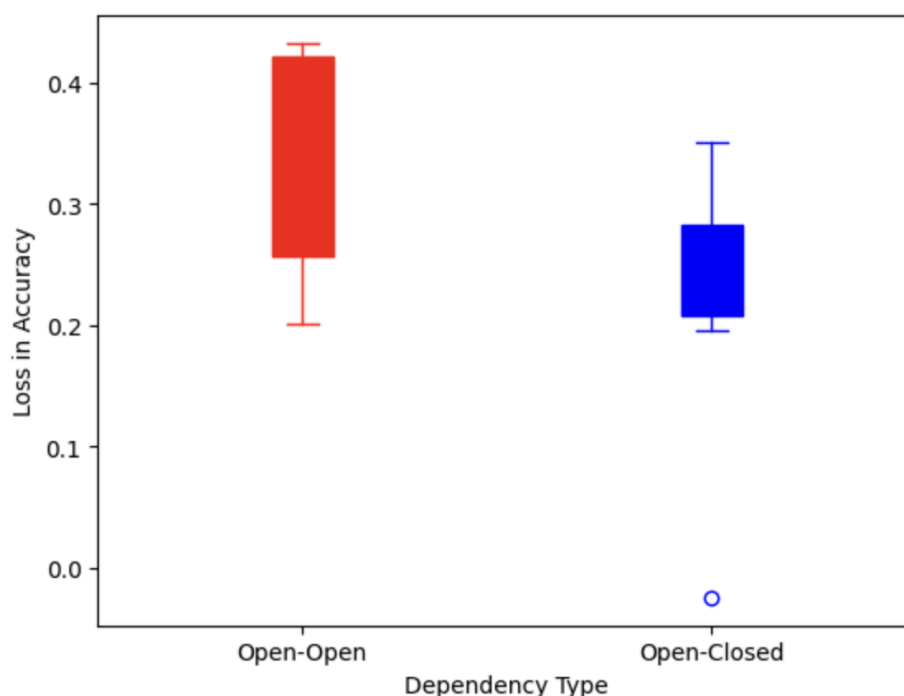


Figure 10: Boxplot of the loss in accuracy for the dependency types (open-open and open-closed). A t-test shows a significant difference in their means ($p=0.041$).

Figure 10 shows the accuracy loss for each dependency type. Open-closed dependencies maintain their accuracy better, and one (punct, connecting words and punctuation) even gains a bit of accuracy. Additionally, open-closed relationships have the highest accuracy after substitution.

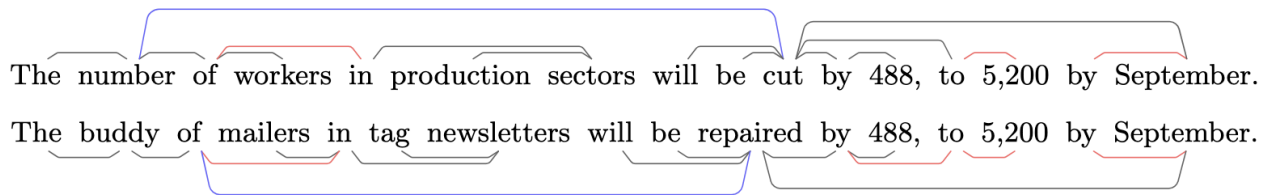


Figure 11: An example of a passive subject dependency (open-open) being mistaken. On top is the probe’s parse of the original sentence, and on bottom is the probe’s parse of the substituted sentence. Here, the blue line on top represents the subject dependency between “number” and “cut”, while the blue line on bottom represents the parser mistakenly placing a dependency between “repaired” and “of”.

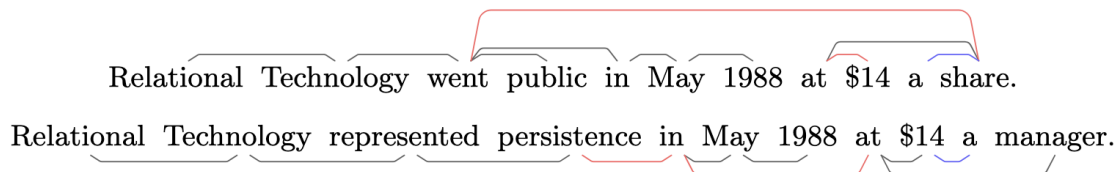


Figure 12: An example of a determiner dependency (closed-open) being mistaken. Here, the probe correctly identifies “a share”, but mistakenly identifies “\$14 a” as the dependency after substitution.

Figures 11 and 12 show some mistakes that the parser made (more mistakes can be found in the Appendix). Frequently, the parser matches a word to a neighbor of the correct word, as is the case in Figure 11 (linking “of” to the main verb rather than “buddy”). Here, it’s possible that the semantic cues simply help to narrow down the area of the correct dependency, but the network may have enough information to know that the initial noun phrase in general should be the subject. Without the semantic information, it chooses a word closer to the center of the noun phrase. Figure 12 shows an example of an error where the chosen word is not a neighbor of the correct answer. When the probe made an error on a closed-open dependency, it frequently attached the word of the closed class to an erroneous word of the open class.

9. Discussion

Our results provide additional evidence against Hewitt & Manning’s claim that their probes are exclusively finding syntax. If they were, then no matter the content of the sentence, two sentences with the same structure should have the same parse. The discrepancy in our UUAS results show that this transformation has some effect on this probe.

To further explore whether our results are based on semantic relations, we check if our errors correlate with semantic dependence. Figures 9-11 show that semantic dependencies tend to do worse under our transformation. However, some clearly grammatical dependencies also do worse; for example, “determiner” is a dependency we had not expected to degrade. However,

overall these demonstrate a general trend of the semantically anomalous substitution causing the probe to treat pairs of words differently, even if they have the same syntactic relationship.

We do not claim that syntax is nowhere to be found in the representational space. However, our results clearly show that relations these probes are exposing are not exclusively syntactic. One potential issue is that the part-of-speech tag associated with a given word is inextricably linked to semantic associations due to the training procedures to get the semantic context in the first place. While the model is not explicitly referencing part of speech information, it is certainly very informative in how one would set up a syntax tree. While an idealized syntactic probe would only have reference to this information, it is hard to disentangle just the grammatical knowledge from the rest of the semantic cloud surrounding that word. Therefore, even a slice of the representation that is primarily about extracting syntax from the input will be related to how much semantic information the model can squeeze out of the word.

This confound is very difficult to remove from experiments looking for syntax in LLMs, and indeed for LLMs as a whole, because LLMs' representations are derived purely distributionally, and empirical distributions of words depend on a combination of syntactic, semantic, pragmatic, and contextual factors. Our results highlight that since semantic information is available to the LLM throughout its training, it is difficult to isolate any other part of the representation.

10. Conclusions

The field of computational linguistics has made massive strides in the past decade. New methods give models more and more information from each data point. Ballooning parameters and training times have allowed them to gain near-complete fluency, and surprising capabilities across multiple tests associated with language understanding.

These same elements that have given LLMs their power have also made their behavior inexplicable. Was this error due to some quirk in the corpus, or some overlooked case in the way it was trained? What set of weights led to the model outputting this word over that one? Why do this model's replies fit better with the surrounding context? The individual layers and pieces that go into building these networks have gotten far too large to look at holistically, and looking at each one individually often gives answers that do not explain very much.

Investigating which generalizations and what elements of underlying structure models detect is therefore a difficult task, but is important as a way to understand how these models are able to process language so well. Hewitt & Manning's structural probes look at the network's word representation space, a very high-dimensional space that encodes the model's "understanding" of the meaning of a word or sentence. The probe searches transformations of this space to look for connections between the words of the input sentence that look like a syntax tree. Hewitt & Manning argue that since these trees are discoverable in the representation space of the model, the model must be using some axis or other part of the space to encode its syntactic knowledge of the input sentence.

This study aimed to fill a gap in this work. While the Hewitt & Manning probe was finding these trees, it did not account for potential semantic relations between these two words, which could also cause them to have meaningful relations within the representation space. A large

proportion of the representation space, and the entirety of its training, is focused on gaining a full semantic view of each word in the corpus so that it can accurately discover the meaning of the sentence. Therefore, any unguided probe that just analyzes the representation space of a network may be affected by latent semantic information. Our methodology removes the semantic relations between the words of the input sentence to see if the probe can still find the same syntax tree, which should be structure-dependent but semantically agnostic.

We found that when we disrupted the semantic relations between the words of our sentences, the probe varied its structure as well. We observed much lower UUAS scores than the probe achieved on its initial dataset. We could also see that different relationships between the words caused the probe to behave differently, even though they seemingly had the same syntactic relationship to each other. Confirming and going beyond the Maudslay & Cotterell results, we have established that the Hewitt & Manning probe is not truly a “syntactic” probe, even when tested with words that are known to the model.

Future work must first investigate ways to corroborate our results, such as investigating if semantic relatedness really does correlate with the number of errors the probe makes. Future work should also look for additional ways to find syntax that are truly indifferent to semantics, which is extremely difficult when the representation space encodes the semantics of the word so deeply. Work like Syntax-BERT (Bai et al., 2021) is a step in the right direction for investigating how models can be made to integrate syntax into their processing.

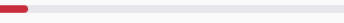
That said, it is unclear whether a model that is better at syntax would actually be a better model for passing context-based benchmarks, or a better model for some non-linguistic downstream task the model will actually be used for. It is important when looking at the results of these models to keep in mind the eventual goals of their creators. People are often not creating models to investigate properties of human language, or aiming to limit themselves to human capabilities when they process language. LLMs today are being wielded as general-purpose tools. Despite active research on explainability and interpretability (Singh et al., 2024; Jia et al., 2025; Balek et al., 2024), our systems may well become more and more inscrutable. Nonetheless, ensuring that our methods are actually finding what we think they are finding is a crucial part of the path to discovering how they work.


11. Acknowledgments

I would like to thank Dr. Resnik and the providers of CCGBank for the dataset I used, as well as Ted Dunning and John Hewitt for providing code that I ran. The data and code I wrote for this project is available upon request. I did not explicitly use any generative AI to write any part of this paper.

Appendix

Example rating question:

0%  100%

 UNIVERSITY OF MARYLAND

These sentences have the same structure:

And she has this inexhaustible energy.

And she covets this common material.

Strongly disagree ☐ ☐ ☐ ☐ ☐ Strongly agree

Mr. Baker drives a 1987 Chevy and usually wears a tweed jacket on his ghostbusting forays.
Mr. Baker stocks a 1987 Chevy and maybe occupies an entrepreneurial complaint on his wine bolts.

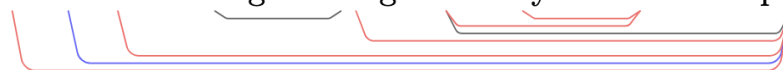
Direct object example

The current transaction cost the bank approximately \$140 million.
The simulated football taught the somatostatin almost \$140 million.

Determiner example

So he 's using river in many project names.

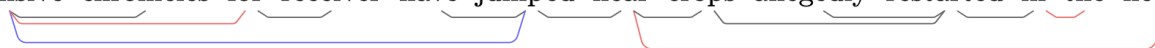
So he 's breeding trading in many standstill repeats.



Nominative subject example

Recent prices for cocoa have been near levels last seen in the mid-1970s.

Expansive chronicles for receiver have jumped near crops allegedly restarted in the heads.



Nominative subject example

Mr. Hall says Mr. Paul was known to spend a lot of money.

Mr. Hall replies Mr. Paul was urged to last a blame of sunrise.



Auxiliary example

Works Cited

- Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., & Goldberg, Y. (2017). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks (arXiv:1608.04207). arXiv. <https://doi.org/10.48550/arXiv.1608.04207>
- Bai, J., Wang, Y., Chen, Y., Yang, Y., Bai, J., Yu, J., & Tong, Y. (2021). Syntax-bert: Improving pre-trained transformers with syntax trees (arXiv:2103.04350). arXiv. <https://doi.org/10.48550/arXiv.2103.04350>
- Balek, V., Sýkora, L., Sklenák, V., & Kliegr, T. (2024). LLM-based feature generation from text for interpretable machine learning (arXiv:2409.07132). arXiv. <https://doi.org/10.48550/arXiv.2409.07132>
- Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., & Roossin, P. (1988). A statistical approach to language translation. Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics. COLING 1988. <https://aclanthology.org/C88-1016>
- Chiang, D. (2007). Hierarchical Phrase-Based Translation. Computational Linguistics, 33(2), 201–228.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for

- statistical machine translation (arXiv:1406.1078). arXiv.
<https://doi.org/10.48550/arXiv.1406.1078>
- Chomsky, N. (1985). Syntactic structures. Mouton Publ. The Hague.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37–46. <https://doi.org/10.1177/001316446002000104>
- Deepseek-AI. (2025, January 21). Deepseek-ai/deepseek-r1 · hugging face.
<https://huggingface.co/deepseek-ai/DeepSeek-R1>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding (arXiv:1810.04805). arXiv.
<https://doi.org/10.48550/arXiv.1810.04805>
- Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1), 61–74. <https://aclanthology.org/J93-1003>
- Futrell, R., Wilcox, E., Morita, T., & Levy, R. (2018). RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency (arXiv:1809.01329). arXiv.
<https://doi.org/10.48550/arXiv.1809.01329>
- Gehring, J., Auli, M., Grangier, D., & Dauphin, Y. N. (2017). A convolutional encoder model for neural machine translation (arXiv:1611.02344). arXiv.
<https://doi.org/10.48550/arXiv.1611.02344>
- Hewitt, J., & Manning, C. D. (2019). A structural probe for finding syntax in word representations. In J. Burstein, C. Doran, & T. Solorio (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)* (pp. 4129–4138). Association for Computational Linguistics.
<https://doi.org/10.18653/v1/N19-1419>
- Hockenmaier, J., & Steedman, M. (2002). Generative models for statistical parsing with combinatory categorial grammar. In P. Isabelle, E. Charniak, & D. Lin (Eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (pp. 335–342). Association for Computational Linguistics.
<https://doi.org/10.3115/1073083.1073139>
- Hockenmaier, Julia, and Mark Steedman. CCGbank LDC2005T13. Web Download. Philadelphia: Linguistic Data Consortium, 2005.
- Honnibal, M., & Montani, I. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*.
- Jia, N., Yuan, C., Wu, Y., & Zheng, Z. (2025). Improving llm interpretability and performance via guided embedding refinement for sequential recommendation (arXiv:2504.11658). arXiv. <https://doi.org/10.48550/arXiv.2504.11658>
- Kasai, J., & Frank, R. (2019). Jabberwocky parsing: Dependency parsing with lexical noise. In G. Jarosz, M. Nelson, B. O'Connor, & J. Pater (Eds.), *Proceedings of the Society for Computation in Linguistics (SCiL) 2019* (pp. 113–123).
<https://doi.org/10.7275/h12q-k754>

- Kim, N., Khilnani, J., Warstadt, A., & Qaddoumi, A. (2023). Reconstruction probing. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 8240–8255). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.findings-acl.523>
- Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 159–174.
- Langsford, S., Stephens, R. G., Dunn, J. C., & Lewis, R. L. (2019). In search of the factors behind naive sentence judgments: A state trace analysis of grammaticality and acceptability ratings. *Frontiers in Psychology*, 10. <https://doi.org/10.3389/fpsyg.2019.02886>
- Lenci, A., & Padó, S. (2022). Editorial: Perspectives for natural language processing between AI, linguistics and cognitive science. *Frontiers in Artificial Intelligence*, 5, 1059998. <https://doi.org/10.3389/frai.2022.1059998>
- Linzen, T., Dupoux, E., & Goldberg, Y. (2016). Assessing the ability of lstms to learn syntax-sensitive dependencies (arXiv:1611.01368). arXiv. <https://doi.org/10.48550/arXiv.1611.01368>
- Maudslay, R., & Cotterell, R. (2021). Do Syntactic Probes Probe Syntax? Experiments with Jabberwocky Probing. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, 124–131.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space (arXiv:1301.3781). arXiv. <https://doi.org/10.48550/arXiv.1301.3781>
- Mysiak, A., & Cyranka, J. (2023). Is German secretly a Slavic language? What BERT probing can tell us about language groups. In J. Piskorski, M. Marcińczuk, P. Nakov, M. Ogrodniczuk, S. Pollak, P. Přibán, P. Rybak, J. Steinberger, & R. Yangarber (Eds.), *Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023)* (pp. 86–93). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.bsnlp-1.11>
- Nivre, J., de Marneffe, M., Ginter, F., Hajič, J., Manning, C., Pyysalo, S., Schuster, S., Tyers, F., and Zeman, D. (2020). Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, pp. 4034–4043, Marseille, France.
- Nuyts, W., Cartuyvels, R., & Moens, M.-F. (2024). Explicitly Representing Syntax Improves Sentence-to-Layout Prediction of Unexpected Situations . *Transactions of the Association for Computational Linguistics*, 2024(12), 262–282.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations (arXiv:1802.05365). arXiv. <https://doi.org/10.48550/arXiv.1802.05365>
- Radford, A., Wu, J., et al. (2019). Openai-community/gpt2 · hugging face. Retrieved April 21, 2025, from <https://huggingface.co/openai-community/gpt2>

- Rastle, K., Harrington, J., & Coltheart, M. (2002). 358,534 nonwords: The arc nonword database. *The Quarterly Journal of Experimental Psychology Section A*, 55(4), 1339–1362. <https://doi.org/10.1080/02724980244000099>
- Singh, C., Inala, J. P., Galley, M., Caruana, R., & Gao, J. (2024). Rethinking interpretability in the era of large language models (arXiv:2402.01761). arXiv. <https://doi.org/10.48550/arXiv.2402.01761>
- Steedman, M. (2000). *The syntactic process*. The MIT Press.
- Tian, Y., Xia, F., & Song, Y. (2024). Large Language Models Are No Longer Shallow Parsers. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 1, 7131–7142.
- Turing, A.M. (1950). Computing machinery and intelligence. *Mind*, 59, 433-460.
- Varanasi, S., Amin, S., & Neumann, G. (2020). Copybert: A unified approach to question generation with self-attention. In T.-H. Wen, A. Celikyilmaz, Z. Yu, A. Papangelis, M. Eric, A. Kumar, I. Casanueva, & R. Shah (Eds.), *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI* (pp. 25–31). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.nlp4convai-1.3>
- Wang, D. P., Sadrzadeh, M., Stanojević, M., Chow, W.-Y., & Breheny, R. (2025). Extracting structure from an LLM - how to improve on surprisal-based models of Human Language Processing. In O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, & S. Schockaert (Eds.), *Proceedings of the 31st International Conference on Computational Linguistics* (pp. 4938–4944). Association for Computational Linguistics. <https://aclanthology.org/2025.coling-main.329/>
- Yamada, K., & Knight, K. (2001). A syntax-based statistical translation model. *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, 523–530. <https://doi.org/10.3115/1073012.1073079>