Neural Networks as Cognitive Models of the Processing of Syntactic Constraints

Suhas Arehalli[1] & Tal Linzen[12]
[1]Johns Hopkins University, [2]New York University

October 2, 2022

Abstract

Languages are governed by *syntactic constraints* — structural rules that determine which sentences are grammatical in the language. In English, one such constraint is *subject-verb agreement*, which dictates that the number of a verb must match the number of its corresponding subject: "the dog*s* run", but "the dog run*s*". While this constraint appears to be simple, in practice speakers make a substantial number of agreement errors, especially if a noun phrase near the verb differs in number from the subject (for example, a speaker might produce the ungrammatical sentence "the key to the cabinets are rusty"). This phenomenon, referred to as *agreement attraction*, is sensitive to a wide range of properties of the sentence; no single existing model is able to generate predictions for the wide variety of materials studied in the human experimental literature. We explore the viability of neural network language models—broad-coverage systems trained to predict the next word in a corpus—as a framework for addressing this limitation. We analyze the agreement errors made by Long Short-Term Memory (LSTM) networks and compare them to those of humans. The models successfully simulate certain results, such as the so-called number asymmetry and the difference between attraction strength in grammatical and ungrammatical sentences, but failed to simulate others, such as the effect of syntactic distance or notional (conceptual) number. We further evaluate networks trained with explicit syntactic supervision, and find that this form of supervision does not always lead to more human-like syntactic behavior. Finally, we show that the corpus used to train a network significantly affects the pattern of agreement errors produced by the network, and discuss the strengths and limitations of neural networks as a tool for understanding human syntactic processing.

Keywords: computational modeling, neural networks, agreement attraction, syntactic processing, psycholinguistics

1 Introduction

Every language is governed by a set of *syntactic constraints* — rules that determine whether a particular sentence is acceptable in that language. These rules are often independent of the meaning of the sentence: Although most listeners would be able to interpret either "the dog **is** running" and "the dog **are** running" as referring to a running dog, only "the dog **is** running" is a grammatical English sentence. A core goal of psycholinguistics is to determine how such syntactic constraints are enforced in real-time sentence production and comprehension.

Amongst those syntactic constraints, *agreement* is both simple and extraordinarily widespread. Put simply, an agreement constraint requires that two or more syntactic elements share a particular set of features. Most varieties of English exhibit *subject-verb number agreement*, where subject noun phrases and their corresponding verbs must share their number feature: They must either both be singular, or both be plural (e.g., "the dog run*s*," but "the dog*s* run").

While this constraint is simple to state, in practice speakers sometimes fail to apply it correctly. Subject-verb agreement errors are particularly likely to arise in sentences with an *attractor*: a non-subject noun phrase with a number feature different than that of the subject (e.g., the attractor "cabinets" might give rise to the erroneous "The key to the cabinets **are** rusty"; Bock & Miller, 1991). These errors occur in both production and comprehension (Bock & Miller, 1991; Pearlmutter, Garnsey, & Bock, 1999), and are

modulated by a number of factors, including, among others, the type of syntactic constituent the attractor appears in (Bock & Cutting, 1992) and the linear or syntactic distance from the attractor to the verb (Vigliocco & Nicol, 1998; Franck, Vigliocco, & Nicol, 2002; Haskell & Macdonald, 2005).

A complete theory of language comprehension and production must provide an account of how syntactic constraints are enforced during processing and of the ways in which the computations enforcing those constraints fail. While many proposals for such an account exist in the literature, no single proposal can account for the full empirical picture, and many make no straightforwardly derivable predictions for much of it. To complicate the situation further, existing models are designed primarily to act as a proof-of-concept, and are typically only able to account for a carefully selected set of materials rather than an arbitrary sentence in the language.

The goal of this paper is to work toward the construction of such a comprehensive account of agreement processing by leveraging the success of the broad-coverage neural network language models—that is, word prediction models—that are widely used in the applied language technologies literature. These language models are designed to take as input a sequence of words and predict the following word in that sequence. They are typically trained on a large corpus of naturally occurring text, which allows them to learn any number of syntactic or semantic properties from their training data, but are provided no explicit supervision, and as such will only learn those properties of the language that are helpful for word prediction. We adopt these models for two reasons. First, unlike previous models of agreement attraction, they are *broad-coverage*: they can take as input any sequence of words and generate predictions for the next word. Second, neural network language models have been shown to be generally capable of enforcing subject-verb agreement in English, but they do make occasional agreement errors (Linzen, Dupoux, & Goldberg, 2016; Gulordava, Bojanowski, Grave, Linzen, & Baroni, 2018). Taken together, these properties allow us to efficiently derive agreement predictions from the models for any set of sentences and compare the errors in those predictions to those made by humans.

Unlike traditional cognitive models, which explicitly implement the mechanisms that researchers hypothesize are used by humans, processing mechanisms in neural language models emerge naturally over the course of training. As a result, it is much more difficult to describe in words the precise cognitive mechanism a neural network model implements. Rather than interpret the exact mechanisms that govern a neural network model's behavior, it is often useful to understand the model in terms of its inductive biases — the types of generalizations the model prefers during learning when multiple generalizations are consistent with its training data. Since the processing mechanisms the model develops over the course of training are the product of the model's inductive biases (as well as the training data itself), inductive biases serve as a simple and useful proxy for reasoning about the mechanisms a neural network learns to use.

We can influence a model's inductive bias by manipulating the model's architecture or its training objective. By comparing the performance of a number of such models, we can characterize the biases that lead a model to produce human-like behavior, which, in turn, allows us to reason about the biases that humans might have. Put another way, this methodology allows us to use the answers for questions about neural networks—i.e., do models with or without a particular inductive bias fail in the same situations as humans?—to inform questions about the mechanisms that humans may use during language processing (i.e., is a particular inductive bias necessary to replicate human error patterns? Or could such an effect emerge from a simpler model?).

We adopt this approach to investigate whether syntactic inductive biases align neural network models more closely with human behavior. We evaluate two types of models based on the Long-Short Term Memory (LSTM) neural network architecture (Hochreiter & Schmidhuber, 1997): models trained solely to predict the next word (LM-Only), and models trained on both word prediction and the syntactic task of Combinatory Categorical Grammar supertagging (LM+CCG, described in Section 2.2). We derive predictions from each of the two types of models for six sets of findings from the human agreement processing literature. Both sets of models successfully simulated a number of empirical findings, but failed to simulate others. Adding the explicit syntactic bias had mixed results: In some cases it aligned the models' error patterns more closely with those of humans, but in other cases it did not.

We then conduct follow-up simulations that aim to identify whether the choice of training data affects the networks' patterns of agreement errors. While in our main experiments, models were trained on a concatenation of a subset of Wikipedia and the CCGBank corpus of news articles (Hockenmaier & Steedman, 2007), in these follow-up experiments we trained models on either the Wikipedia subset or CCGBank. We

found that both the size and genre of the training corpus affected the errors the models made. This suggests that neural networks used as cognitive models may need to incorporate stronger inductive biases, not only to encourage more human-like behavior, but also to reduce sensitivity to the composition of their training corpora. The fact that even small differences in corpus composition can lead to significant differences in model behavior additionally highlights the importance of creating training corpora that accurately reflect the data humans learn from.

All of our LSTM-based models, which were trained on moderately-sized corpora by the standard of the language technologies world, displayed large average error rates relative to humans. To investigate whether this reflects an issue with neural network models more generally, we conducted an additional follow-up simulation using the publicly available GPT-2 language model (Radford et al., 2019), which was trained on many billions of words and is based on the Transformer neural network architecture (Vaswani et al., 2017). We found that, though GPT-2 displays a lower overall error rate, this overall improvement does not translate into a more human-like error patterns.

Before we describe our simulations in detail, we provide a brief introduction to agreement and agreement attraction in English, as well as briefly discuss related prior work modeling human language processing with neural language models and how the present work fits into this landscape.

1.1 Subject-verb agreement and agreement attraction in English

Subject-Verb agreement is a constraint in many dialects of English that requires the number feature of a subject to match the number of the corresponding verb, as in Example 1. A mismatch in number features results in the ungrammatical Example 2.

(1) The key opens the door.

(2) *The key open the door.

This constraint holds regardless of what noun phrases (NPs) appear elsewhere in the sentence, as shown in Example 3 and Example 4.

(3) The key to the cabinet opens/*open the door.

(4) The key to the cabinets opens/*open the door.

In practice, human behavior can deviates from this description. Agreement errors occur occasionally in many contexts, and are particularly common in the presence of a non-subject NP whose number feature does not match that of the subject, such as Example 4: In this example, a higher error rate is expected compared to the minimally different Example 3 (Bock & Miller, 1991).

This pattern of errors was originally documented in the sentence completion paradigm. In this paradigm, participants are given a prefix of a sentence up to but not including the main verb, as in Example 5 or 6, and are tasked with completing the sentence:

(5) The key to the cabinets...

(6) The key to the cabinet...

The experimenter then determines if the participant produced a grammatical verb that matches the number of the subject, like "is", or an ungrammatical verb, like "are". Following Bock and Miller's seminal study, agreement attraction has also been documented in comprehension (Pearlmutter et al., 1999; Parker & An, 2018; Wagers, Lau, & Phillips, 2009), as we discuss in Section 2.3.1, and similar findings have been reported across languages (Lorimor, Bock, Zalkind, Sheyman, & Beard, 2008; Franck et al., 2002; Franck, Lassi, Frauenfelder, & Rizzi, 2006, among others)

The magnitude of the agreement attraction effect—the difference in error rates between Example 5 and 6, for example—is sensitive to a variety of factors, both syntactic (Bock & Cutting, 1992; Franck et al., 2002, inter alia) and semantic (Humphreys & Bock, 2005; Parker & An, 2018, inter alia). A number of theories have been proposed to explain the influence of these factors on agreement; these include the Marking & Morphing model (Eberhard, Cutting, & Bock, 2005, etc.), feature percolation accounts (Franck et al., 2002, etc.), and memory retrieval-based accounts (Wagers et al., 2009, etc.). Each account is motivated by a particular subset of the empirical findings that are best explained by that account: Notional number effects motivate

the Marking & Morphing model (Humphreys & Bock, 2005, etc.), syntactic distance effects motivate feature percolation accounts (Bock & Cutting, 1992; Franck et al., 2002, etc.), and linear distance effects (e.g., Haskell & Macdonald, 2005) and grammaticality asymmetry effects (Wagers et al., 2009) motivate memory retrieval-based models.

In this paper, we use neural networks to simulate six human experiments that span the three subsets of results that have motivated previous accounts. The findings of these experiments can be summarized as follows: (1) Attractors in prepositional phrases give rise to a stronger attraction effect than those in relative clauses, and plural attractors generate a stronger attraction effect than singular attractors (Bock & Cutting, 1992); (2-3) Attractors closer to the verb exert a stronger attraction effect, whether distance is measured in syntactic (Franck et al., 2002) or linear (Haskell & Macdonald, 2005) terms; (4) Collective subjects with distributive readings have higher rates of plural agreement than those with collective readings (Humphreys & Bock, 2005); (5) Attractors in oblique arguments cause a larger attraction effect than those in core arguments (Parker & An, 2018); and (6) Attraction can be caused by attractors outside of the clause containing the agreement dependency, and while attraction makes ungrammatical sentences seem grammatical, it does not make grammatical sentences seem ungrammatical (Wagers et al., 2009).

1.2 Subject-verb Agreement in Neural Language Models

Most relevant prior work has evaluated the extent to which neural networks obey agreement constraints, and was not directly concerned with modeling human agreement error patterns. Elman (1991) evaluated Simple Recurrent Networks (SRNs) trained to predict the next word in a small artificial corpus and found that the models were capable of predicting the number of verbs accurately, even when the subject and verb were separated by a relative clause. More recently, Linzen et al. (2016) trained Long-Short Term Memory models (LSTMs) using a number of objectives, including word prediction, and evaluated whether they predicted the correct number inflection of the verb on preambles extracted from Wikipedia, which include naturally occurring attractors. While they concluded that word prediction alone was insufficient to learn agreement dependencies from natural corpora, Gulordava et al. (2018) later reached a different conclusion, demonstrating that a better trained LSTM language model could successfully learn agreement dependencies through word prediction, even when evaluated on so-called "colorless green ideas" preambles that are stripped of any potentially useful semantic content. Agreement across simple intervening noun phrases has also been a consistent part of syntactic benchmarks for language models (Marvin & Linzen, 2018; Warstadt, Singh, & Bowman, 2019; Warstadt et al., 2020; Hu, Gauthier, Qian, Wilcox, & Levy, 2020), with modern models performing reasonably well, though with some errors.

Taken together, this body of work provides robust evidence that neural network language models are capable of representing subject-verb number agreement dependencies, though these representations have their limitations. *What* representations those models employ for this agreement, and how robust those representations are, on the other hand, is much less clear. One line of work aiming to address this question has found evidence for single units in RNN-based models that represent number information for all subject-verb relationships within a sentence (Lakretz et al., 2019, 2021). Another line of work analyzing GPT-2 (Radford et al., 2019), a Transformer-based model (Vaswani et al., 2017), suggests that attraction effects may be the result of a part of the transformer architecture being subject to the same sorts of similarity-based interference effects as cue-based models from the human memory literature (Ryu & Lewis, 2021).

Most prior work has not compared the neural networks' detailed error patterns to those of humans. One exception is Linzen and Leonard (2018), who found that the models they trained exhibited agreement attraction errors, in general, as well as number asymmetry effects (with plural noun phrases exerting a stronger attraction effects than singular ones), but did not show higher error rates with attractors in prepositional phrases than with attractors in relative clauses (as was found for humans by Bock & Cutting, 1992). However, the models used by Linzen and Leonard (2018) were not word prediction models, but classifiers trained solely to predict the number feature of the verb.

Like Linzen and Leonard (2018), the current work aims to model the patterns of agreement errors that humans produce, but we use models trained on the general, broad-coverage word prediction task, rather than models tailor-made for the agreement task. This requires us to use linking functions that relate the probability distribution over the upcoming word defined by the model, on the one hand, and human behavioral measures on the other hand. In our simulations of production studies, we use a probabilistic linking hypothesis between

our language model's probabilities over the identity of a verb and the number feature our model predicts; this linking function better captures the model's confidence in its prediction. Second, to model processing difficulty in comprehension-based studies of agreement attraction, we adopt surprisal-based methodologies employed by work that investigates processing phenomena typically studied through online comprehension measures (i.e., garden paths, filler-gap dependencies, etc.; van Schijndel & Linzen, 2018; E. Wilcox, Levy, Morita, & Futrell, 2018; Futrell et al., 2019; van Schijndel & Linzen, 2021, among others). We discuss these linking hypotheses, as well as our modeling and statistical choices, in detail in the next section.

2 Methods

2.1 Language Models

Language models are systems that predict the next word in a sequence. We primarily use language models based on the LSTM architecture, a type of *Recurrent Neural Network* (RNN) architecture. We briefly describe this neural network architecture in the remainder of this section.

RNNs transform a sequence of vector representations (representing, for example, words in a sentence) into a single vector representation by iteratively merging a vector representation of the left context (h_{i-1}) with a vector representation of the input to the right of that context (w_i) until all of the vectors are merged. In Simple Recurrent Networks (Elman, 1990, SRNs), vectors are merged using Equation 1. The weight matrices W_h and W_w are learned linear transformations that are applied to h_{i-1} and w_i respectively; the outcomes are summed and transformed by a non-linear activation function (in this case, the hyperbolic tangent function):

$$h_i = \tanh(W_h h_{i-1} + W_w w_i) \tag{1}$$

In a neural network language model, words from the training data are mapped to learned vector embeddings, and sequences of those embeddings are fed into a neural network encoder that, like the recurrent network described above, produces a single vector that represents that sequence of words. That representation is then provided as the input to a linear decoder — a learned linear transformation followed by a softmax operation — which outputs a probability distribution over the model's vocabulary (see Figure 1). The model's task is to align this probability distribution with the empirical probability that any particular word in the model's vocabulary is the next word in the sequence. Before training, all of the model's learned weights — in a simple recurrent network, those are the embedding mappings, the two weight matrices W_h and W_w, and the matrix representing the linear transformation in the encoder — are randomly initialized, and so the model's output probability distribution is essentially random. For each training example, all of those weights are adjusted using stochastic gradient descent so as to increase the likelihood of the true next word from the training data.

Our simulations primarily use LSTMs, a type of RNN that incorporates gating mechanisms designed to maintain representations over longer sequences; these mechanisms mitigate the issue that, due to successive merging operations, representations derived from early words have little effect by the end of the sequence. These gating mechanisms yield better representations of long-distance dependencies (Bhatt, Bansal, Singh, & Agarwal, 2020), which makes them better suited than SRNs for modeling agreement relations, and, in turn, agreement attraction. On a conceptual level, however, LSTMs fundamentally operate by the same principles as SRNs: they incrementally merge inputs from left to right using a trainable, parametrized function.

In order to evaluate whether more sophisticated model architectures and training regimes can address issues of high error rates found in our LSTM-based models, we additionally consider GPT-2, a large Transformer-based langauge model. Unlike the RNN models described above, Transformer-based language models do not predict the next word from a representation generated by an incremental left-to-right composition operation. Instead, transformers construct operations using a mechanism called *self-attention*, where the representation used to predict the next word is composed with with direct access to the representations of all prior words within a large context window.

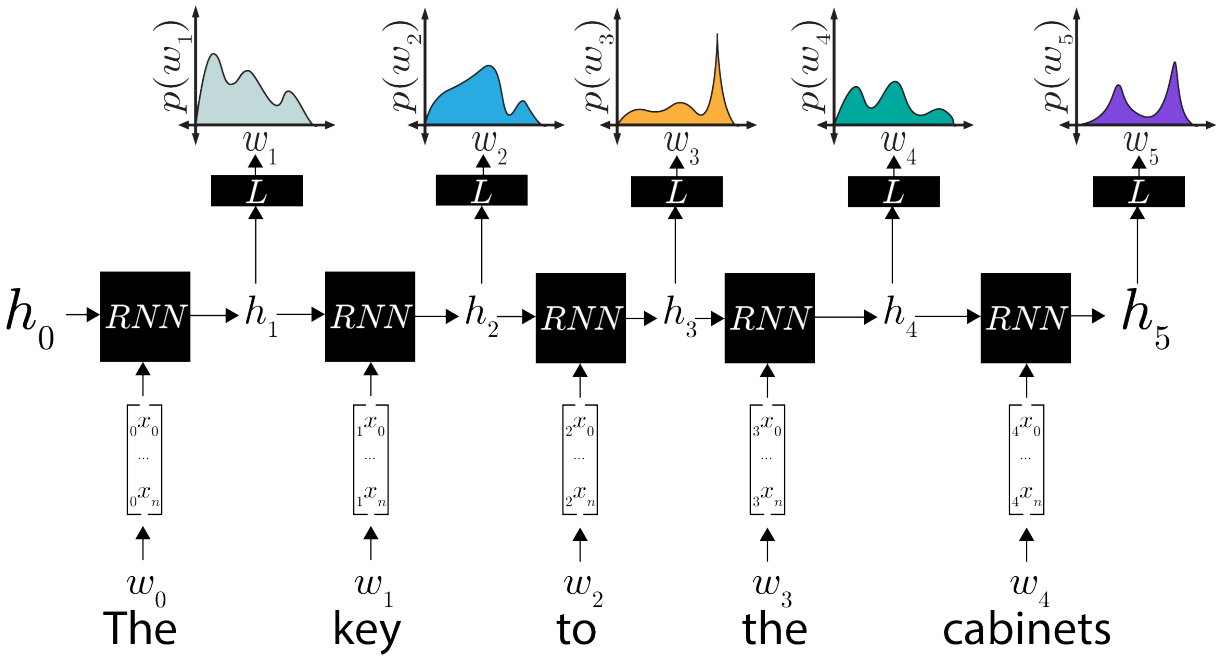


Figure 1: In our language modeling setup, each word is mapped to a word vector. Each of those representations is combined with a representation of all previous words (h_{i-1}) using a recurrent neural network model (RNN) to create a representation for all words up to and including word i, h_i. To generate a prediction for word $i+1$, that word's representation, h_i is fed into a linear decoder (L) to generate a distribution over word $i+1$. During training, model weights (which determine RNN and L) are adjusted to maximize the probability of the word that actually occurred in the sentence at position $i+1$.

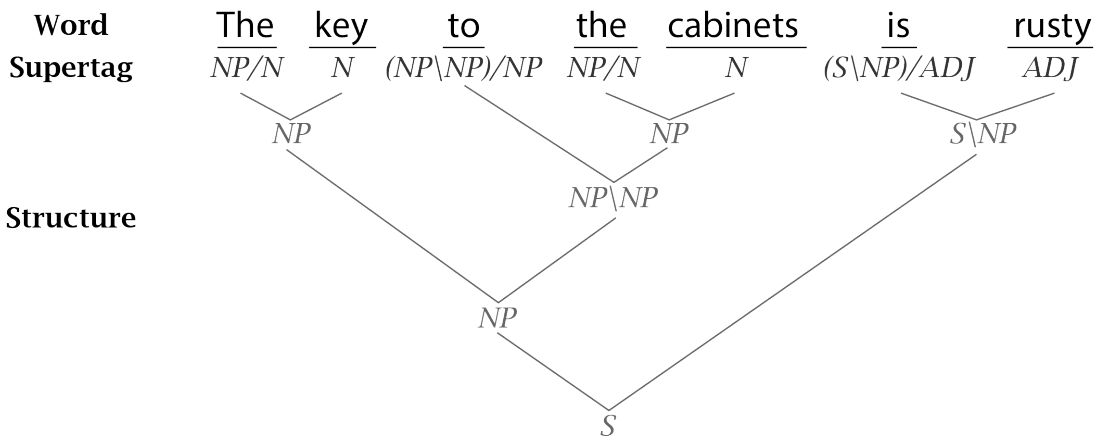


Figure 2: An example sequence of CCG supertags for the sentence *The key to the cabinets is rusty*. Each supertag encodes how the corresponding word composes with its syntactic neighborhood. The label Y/X denotes that the word it labels merges with a constituent of type X on its right to form a constituent of type Y (as with *the* and *key*), and $Y\backslash X$ denotes the same, but with the constituent of type X on its left (as with *to the cabinets* and *the key*). To predict supertags successfully, models must learn to represent something akin to the underlying structure of the sentence. In many cases, knowing the sequence of supertags making it possible to deterministically reconstruct the full parse of the sentence.

2.2 Model Architectures and Training Setup

For each of the six humans experiments we discuss, we compare human behavior to simulation results for two types of models: models trained only on word prediction (LM-Only models), as described in the previous section; and multi-task models, which are trained on both word prediction and *Combinatory Categorial Grammar Supertagging* (LM+CCG). The multi-task models are trained to predict, from a sequence of words, not only the next word, but also the most recent word's *supertag*—an enriched part-of-speech tag that encodes local syntactic information (see Figure 2). Due to the rich syntactic information contained in supertags, supertagging has been described as "almost parsing" (Bangalore & Joshi, 1999), and so we hypothesize that jointly optimizing for both supertagging and language modeling accuracy will imbue a model with an additional bias toward learning more sophisticated syntactic representations (Enguehard, Goldberg, & Linzen, 2017).

We trained five instances of each model. The weights of each of these instances was randomly initialized separately; training multiple model instance with different initial weights allows us to determine to what extent the behavior observed is dependent on particular initial weights (McCoy, Min, & Linzen, 2020), much like group-level analyses in psychology. The five LM-Only model instances were trained for 12 epochs over the 80 million words of English Wikipedia used in Gulordava et al. (2018), concatenated with the approximately one million words of the Wall Street Journal section of the Penn Treebank (Marcus, Santorini, & Marcinkiewicz, 1993). Following Gulordava et al. (2018), the RNN encoder in each model was a 2-layer LSTM with 650 hidden units in each layer. LM-Only models achieved perplexities between 66.73 and 67.13 over the Wikipedia corpus' test set.

The five LM+CCG model instances were trained on both word prediction and supertagging: In addition to the linear decoder that predicted the next word, a secondary linear decoder predicted the current word's supertag. The structure of this multi-classifier architecture is outlined in Figure 3. The supertags we used are derived from the Combinatory Categorial Grammar (CCG) formalism (Steedman, 1987). Word prediction was performed over the 80 million words taken from English Wikipedia (Gulordava et al., 2018), supplemented with approximately one million words of the Wall Street Journal section of the Penn Treebank. CCG supertagging was performed over CCGbank (Hockenmaier & Steedman, 2007), a version of the Penn Treebank Corpus annotated with CCG derivations. The two training objectives—word prediction and supertagging—was weighted equally in training. LM+CCG models achieved language modeling perplexities ranging from 74.76 to 75.70 on the Wikipedia test set, and assigned the highest likelihood to the correct CCG

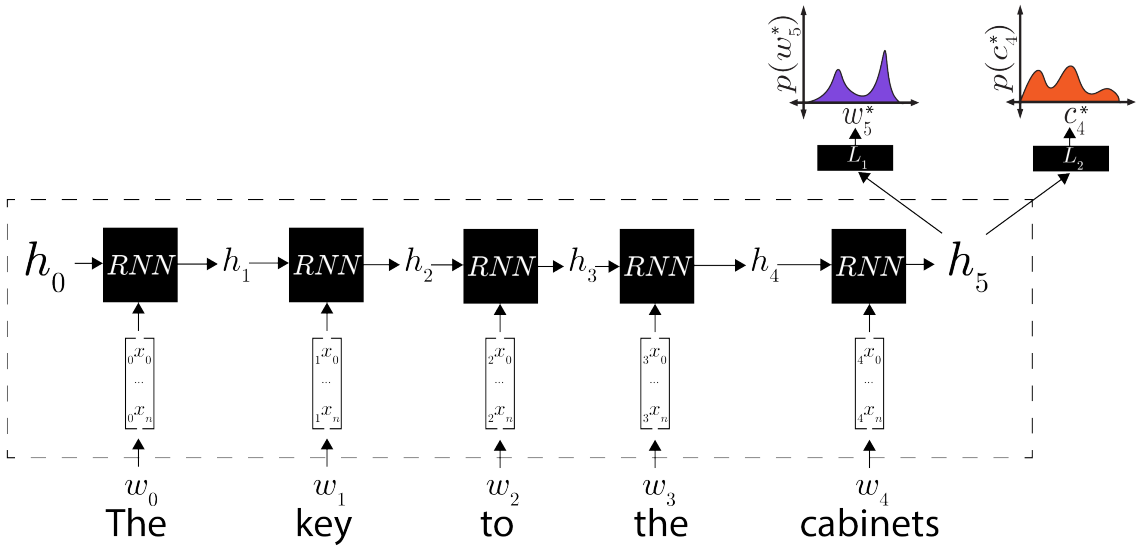


Figure 3: An outline of the architecture used for the LM+CCG models. Using the internal representation h_5 constructed by an RNN encoder, classifier L_1 predicts the next word and classifier L_2 predicts the current word's supertag.

supertag between 84.1% and 84.5% of the time. This is substantially higher than the accuracy of a model that selects the most frequent supertag for each word independent of the context, which is 71.2% (Clark, 2002); this suggests that the models have learned a considerable amount about local syntactic structure, and thus lends credence to our belief that supertagging imbues our model with stronger syntactic inductive biases.

The models described so far were trained on the concatenation of two distinct corpora. To assess the contribution of each corpus, we trained two additional sets of models with the LM-Only architecture on each of those corpora: five model instances on the 80 million words of Wikipedia, and five on the approximately one million words of the Wall Street Journal section of the Penn Treebank.[1] Test-set perplexities for models trained on Wikipedia data were in the 67.66–68.15 range, and those for models trained on Penn Treebank data were in the 55.32–56.13 range. Note that since these perplexities are computed over the test set corresponding to each model's training set, the perplexities for the two sets of models are not comparable.

Finally, evaluations using GPT-2 employed the 117 million parameter GPT-2 model (Radford et al., 2019), trained on roughly 40GB of text scraped from the internet. This model achieves a perplexity of 65.85 over the Penn Treebank.[2]

2.3 Linking model outputs to human behavior

The behavioral data in the experiments we simulate has one of two forms: the proportion of singular verbs produced in a sentence completion paradigm, or the reading time of words in a critical region in a self-paced reading study. Both paradigms are discussed in more detail in this section. As we described in Section 2.1, a language model takes as input a sequence of words and outputs a probability distribution over the next word in that sequence. To compare the performance of these models to that of humans, we need to link

[1]Note that while the analyses we present here are similar to those presented in Arehalli and Linzen (2020), the LM-Only models trained over Wikipedia we use are different. While Arehalli and Linzen (2020) train instances of the exact model architecture used in Gulordava et al. (2018), the models trained here are roughly matched to LM+CCG models in training time and training hyperparameters to allow for straightforward comparisons between the two types of model.

[2]Note that since GPT-2's vocabulary consists of sub-word tokens rather than full words, perplexities are not directly comparable to our other models trained on the PTB, which predict whole words.

the language model's output to the behavioral responses recorded in the human experiments. This section discusses how we select an appropriate linking function, and how we combine it with a language model to construct what we will, in future sections, refer to simply as our (cognitive) model.[3]

2.3.1 Predicting reading times

The comprehension studies we simulate have employed the self-paced reading paradigm. In self-paced reading, participants are presented with sentences one word at a time; the next word is revealed after the participant presses a particular button. The dependent measure is the time that elapses between two key presses (the displayed word's *reading time*). Longer reading times are taken to indicate greater processing difficulty caused by the word currently being displayed, or by one of the words immediately preceding it.

In the context of agreement processing, reading times at the verb can indicate how acceptable the participant finds the subject-verb agreement relation in question. The logic of this paradigm relies on the observation that encountering an agreement violation incurs processing cost, which leads to longer reading times at the verb or at the words immediately after it. Agreement attraction can then surface in one of two manners: The amelioration of an agreement error, where ungrammatical sentences are read faster when an attractor matches the number of the verb; and the illusion of an agreement error, where grammatical sentences are read slower when an attractor mismatches the number of both the subject and verb (Pearlmutter et al., 1999; Wagers et al., 2009). We will discuss this logic in more detail when we describe the two comprehension experiments we simulate (see Sections 3.5 and 3.6).

For each word in the input sentence, language models provide us with a probability distribution over the vocabulary, which represents the model's predictions for the next word. To convert these probability distributions into a measure comparable with reading times, we use *surprisal* (Hale, 2001; Levy, 2008), defined in Equation 2.

$$\mathit{Surprisal}(w_i) = -\log_2(P(w_i \mid w_0, ..., w_{i-1})) \tag{2}$$

Note that the probability $P(w_i \mid w_0, ..., w_{i-1})$ is the probability that the i-th word in the sequence is w_i, given that all of the prior words are $w_0, ..., w_{i-1}$. This is precisely the probability distribution we obtain from a language model after it has been given $w_0, ..., w_{i-1}$ as input. The relationship between human reading times and surprisal estimated from a language model in this fashion has been found to be approximately linear (Smith & Levy, 2013).

2.3.2 Predicting verb completions

The production studies we simulate all used the sentence completion paradigm briefly described above. In this paradigm, participants are asked to repeat and complete a given preamble (in this case, a complex noun phrase), and their responses are coded for the number feature of the verb they produce and whether the agreement relation is grammatical. For example, when provided the preamble "The keys to the cabinet", a participant might respond with "The keys to the cabinet are on the table", which would be coded as a plural, and in this case grammatical, response. Agreement attraction manifests in a higher error rate for preambles where the attractor noun's number mismatches the subject's number compared to preambles where the attractor noun's number matches the number of the subject. To simulate such an experiment with language models, we need to convert the output of the language model — a distribution over the next word in the sentence — to the probabilities with which the model would produce a singular or plural verb.

For our simulations, we will use what we will refer to as the ONE-SAMPLE linking function. This function is equivalent to having the simulated production process decide on a verb form based on a single sample from the underlying language model's probability distribution (see Section 4.3 for more details and the motivation for the name ONE-SAMPLE). Under this paradigm, we first select a candidate pair of singular and

[3]We use the term "cognitive model" here only to distinguish the models we create, which aim to predict human experimental measures like error rates and reading times, from the language models that underlie them, which aim only to predict the next word. While our eventual goal is to use such models to investigate the cognitive processes that generate those experimental measures, we do not use the term here to indicate that these models provide an explicit, interpretable account of a particular human cognitive process. See Section 4.4 for a further discussion of how these models relate to the more traditional cognitive models used in psycholinguistics.

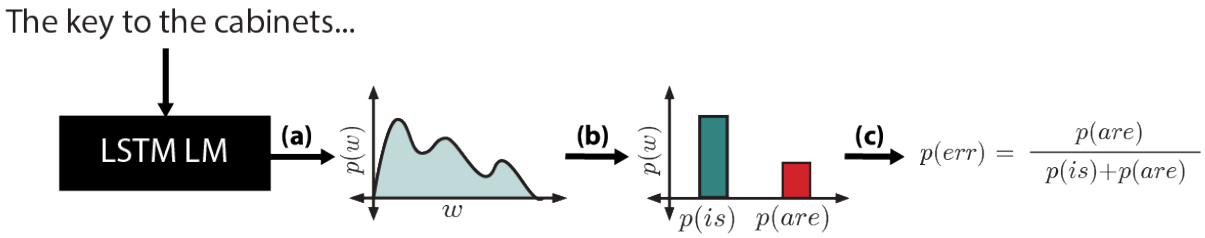


Figure 4: To simulate a sentence completion experiment, a language model is given each preamble as input, producing a probability distribution over the following word (***a***). The probabilities of a candidate singular and plural verb are extracted from this distribution (***b***) and renormalized (***c***) and this new distribution is taken to represent the probability with which the model would produce a singular or plural verb.

plural forms of a particular verb — for example, *is* and *are* — and compute their probabilities under the distribution provided by the language model. We then renormalize the probabilities over the two candidate words such that they sum to 1, and take the renormalized probabilities as the probabilities with which the model produces a singular or plural verb (see Figure 4).

2.4 Experimental Stimuli

For each simulation, we aimed to use the stimuli provided in the publications that reported on the relevant human experiment. This goal was complicated by the fact that the models can only process words included in their training data; some of the more infrequent words in the experimental stimuli did not occur in the training corpus at all, or, following standard practice, were replaced during training with a standard "unknown" (out-of-vocabulary) token.[4] To deal with this issue, we identified any out-of-vocabulary word that was a part of a noun phrase (and thus could potentially contribute number information) or was manipulated in the simulated experiment's design and replaced it with a semantically similar, in-vocabulary word. Note that this necessarily increases the frequency of the word as estimated using our training corpora, since the original words did not appear in the models' vocabularies—precisely because it fell under the out-of-vocabulary frequency threshold—while the replacement word did appear in the vocabulary. If the word was not in a noun phrase, or was not relevant to the experimental manipulation, we did not attempt to find a substitute word, and replaced it with the out-of-vocabulary token instead. A summary of the changes we made to the materials can be found in Appendix A.1. Due to the limited vocabulary of the models trained on the Penn Treebank, a larger number of words needed to be adjusted. To avoid editing experimental materials too significantly, while still comparing models trained on the two datasets, we limited our simulations based on these models to the three experiments that focused on syntactic structure: Bock and Cutting (1992), Franck et al. (2002), and Haskell and Macdonald (2005). The relevant changes to the materials are listed in Appendix A.2.

The candidate pairs of singular and plural verbs for production experiments were always forms of the verb *be*. We did this for two reasons: first, these verbs appear with high frequency in the training data, and thus are likely to have number information properly encoded in their vector representations; and second, these verbs are plausible with nearly any subject noun phrase, and thus can be used across a wide variety of stimuli.

2.5 Statistical Analysis

For each of our statistical analyses, we first constructed a mixed-effects model with a maximal mixed-effects structure, that is, random slopes and intercepts for each experimental item and model instance. If the statistical model did not converge, the random effects structure was incrementally pruned until convergence was reached. For all mixed-effects models reported below, this procedure resulted in the inclusion of random intercepts only, for both items and model instance.

[4]This is done typically done to account for the fact that language models are unable to learn appropriate vector representations for words that occur a small number of times in the training corpus.

For the analyses where the response variable was surprisal, we used linear mixed-effects regression. For the analyses where the response variable was a probability, we used beta mixed-effects regression (Ferrari & Cribari-Neto, 2004), which assumes that the dependent variable (the probability of a particular inflection of the verb) is beta distributed. This assumption bounds the value of the dependent variable between 0 and 1, as is appropriate for a probability. To test the significance of each fixed effect, we report the result of either a Wald test (for beta mixed effects models) or a t-test (for linear mixed effects models). To test whether two fixed effects are significantly different from each other, we report the results of a linear hypothesis test where we compare the fit of the mixed-effects model to a model where the two fixed effects in question are constrained to be equal.

3 Simulations

This section describes the results of simulations of the six experiments from the human literature that we examine in this paper. For each experiment, we lay out the motivation and design of the experiment, describe the outcome of the human experiment, and report the results of our simulations. In Section 3.7, we synthesize the results of the simulations with respect to the three empirical questions we seek to answer: (1) What agreement phenomena do LM-Only language models capture? (2) What effect does the addition of an explicit syntactic training objective have on a model's agreement behavior? (3) How does a model's agreement behavior depend on the corpus used to train the model?

3.1 Attractors in prepositional phrase vs. relative clauses

Background: The first three experiments we simulate investigate how hierarchical syntactic structure affects agreement attraction. We first simulate Experiment 1 of Bock and Cutting (1992), which sought to determine whether the syntactic environment in which an attractor appears affects the strength of attraction. In particular, the authors tested whether attractors located within prepositional phrases (PPs, Examples 7–8) exerted a stronger attraction effects than attractors within relative clauses (RCs, Examples 9–10):

(7) The demo tape from the popular rock singer...

(8) The demo tape from the popular rock singers...

(9) The demo tape that promoted the rock singer...

(10) The demo tape that promoted the rock singers...

Human results: Using the sentence completion paradigm (see section 2.3.2), Bock and Cutting compared the strength of the attraction effect within PPs (the difference in error rates between preambles like Example 7 and 8) to that within RCs (the difference in error rates between Example 9 and 10). They found that attraction was stronger from attractors in PPs than attractors within RCs. They also documented a *number asymmetry*: there were more attraction errors in sentences with singular subjects than in sentences with plural subjects.

Simulation results—modifier type: A comparison of the human results and simulations using LM-only and LM+CCG models is shown in Figure 5. Both types of models exhibited a significant attraction effect (LM-Only: $\beta = 0.91$, $|z| = 34.19$, $p < 0.001$; LM+CCG: $\beta = 0.78$, $|z| = 24.14$, $p < 0.001$). However, unlike humans, LM-Only models exhibited no interaction between the attraction effect and the type of modifier the attractor appeared in ($\beta = -0.017$, $|z| = 0.66$, $p = 0.51$), and the LM+CCG models showed no significant interaction ($\beta = -0.058$, $|z| = -1.18$, $p = 0.07$). The three-way interaction between attraction, syntactic environment (PP vs. RC), and model type (LM-only vs. LM+CCG) found no evidence for any difference in the performance of the two types of models ($\beta = 0.041$, $|z| = 1.00$, $p < 0.31$). In summary, neither type of model successfully simulated the human pattern.

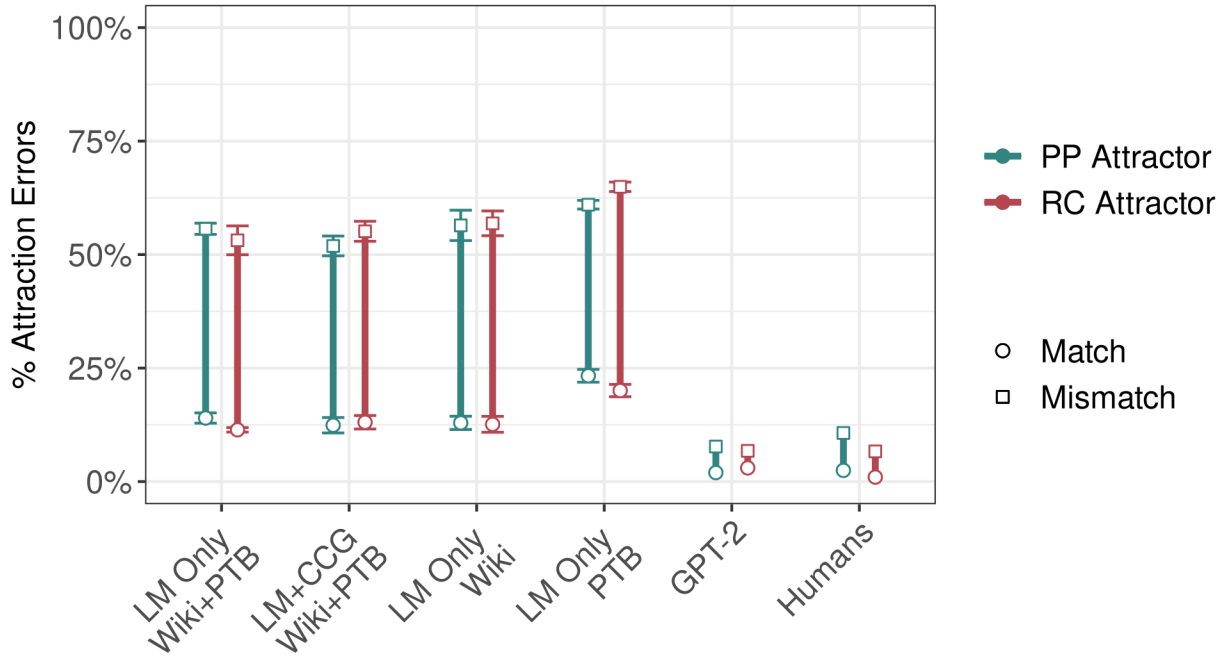


Figure 5: Human and simulation results for Bock and Cutting (1992). Vertical bars represent the size of the attraction effect: the difference between the subject-attractor number match condition (the lower, circular endpoints) and mismatch condition (the higher, square endpoints). Error bars represent standard errors across the five randomly initialized models trained for each model architecture and training set. If the models simulate the relevant result from Bock and Cutting (1992), the attraction effect in RCs (the length of the red bar) is smaller than that in PPs (The length of the green bar). This pattern is reversed in LM-Only models trained on the Penn Treebank, and no significant difference is found between modifier types in all other models.

Simulation results—number asymmetry: Simulations using both models replicated the number asymmetry (LM-ONLY: $\beta = 0.20$, $|z| = 5.47$, $p < 0.001$; LM+CCG: $\beta = 0.34$, $|z| = 7.40$, $p < 0.001$). There was a significant 3-way interaction between attraction, subject number, and model type ($\beta = -0.16$, $|z| = 2.66$, $p < 0.01$), with LM+CCG exhibiting greater number asymmetry than LM-ONLY. In contrast to the effect of modifier type, then, the number asymmetry effect was captured by both types of models and was stronger in LM+CCG models.

Sensitivity to training corpus: LM-ONLY models trained on the smaller Penn Treebank dataset displayed a significant attraction effect ($\beta = 0.85$, $p < 0.001$, $|z| = 24.14$), and an interaction between the attraction effect and the type of modifier ($\beta = -0.09$, $p < 0.01$, $|z| = 2.63$), such that attractors led to more errors when they were in relatives clauses than when they were in prepositional phrases. This effect was, crucially, in the opposite direction of that found in humans. Models trained on the larger Wikipedia dataset also exhibited an attraction effect ($\beta = 0.94$, $p < 0.001$, $|z| = 8.32$) but no interaction between that effect and modifier type ($\beta = 0.0084$, $p = 0.76$, $|z| = 0.31$). The Wikipedia-trained models exhibited a number asymmetry ($\beta = 0.22$, $p < 0.001$, $|z| = 5.60$), while Penn Treebank-trained models did not ($\beta = 0.053$, $|z| = 1.08$, $p = 0.28$). The two types of models differed in the magnitude of the interaction between attraction and type of modifier was significant, as assessed by a three-way interaction ($\beta = 0.15$, $|z| = 2.29$, $p < 0.05$); this was also the case for the analogous three-way interaction between model type, attraction and number ($\beta = 0.10$, $|z| = 2.31$, $p < 0.05$).

This pattern of results suggests a strong influence of dataset on the ability to replicate the difference in error rates between attractors in PPs and RCs, even with no difference in model architecture or training objective. While models trained on the smaller Penn Treebank dataset produced the wrong verb more often when the attractor was in an RC, models trained on the larger Wikipedia dataset showed no difference in error rates between the two conditions. While neither matched human behavior—more errors when attractors appear in PPs compared to RCs—training on Wikipedia resulted in more human-like results than training on the Penn Treebank.

Overall agreement error rate: Human error rates, even in the conditions in which error rates were highest, were less than 15%. By contrast, models routinely made agreement errors in more than 50% of trials when an attractor was present. Though this difference in magnitude indicates that the models we trained are particularly susceptible to attraction errors, we take this discrepancy to be largely orthogonal to the goals of our investigation. We are concerned primarily with (1) whether our simple models exhibit agreement attraction (which high rates of agreement errors makes apparent), (2) whether the factors we investigate modulate error rates in the same way in humans and models, and (3) whether changes to models, either in training data or training objective, lead to more human-like behavior. Since these motivating questions consider only how differences in error rates change across various conditions, we have no reason to believe that high overall error rates are problematic for our analyses.

We must, however, consider the ways in which the changes we could make to reduce the overall error rate could affect differences in in error rates across conditions. In particular, changes we make to reduce the overall error rate could imbue models with inductive biases that affect the kinds of errors our models make. For instance, we select LSTM language models because they lack strong biases toward sophisticated syntactic representations. Since sophisticated syntactic representations are key to identifying the subject and avoiding agreement errors, the high rate of errors is tied directly to our choice of an unbiased model for this evaluation. Though adjusting the base rates of errors is not our goal in this work, we present potential avenues for doing so in Section 3.7.2.

Will a lower overall error rate lead to greater alignment with the human error pattern? While a higher overall error rate is not unexpected in our intentionally simple and unbiased LSTM models, it would be problematic to our approach if the higher agreement error rate was not tied to the weakness of a specific model, but reflected a limitation of neural network models broadly. To address this concern, we analyze the results of our simulations with GPT-2, a much larger model based on the Transformer architecture (as opposed to the RNN-based architecture that underlies our models). Overall, GPT-2 error rates are largely comparable to human error rates in all conditions (ranging between 1.198% and 7.72%). GPT-2 exhibited

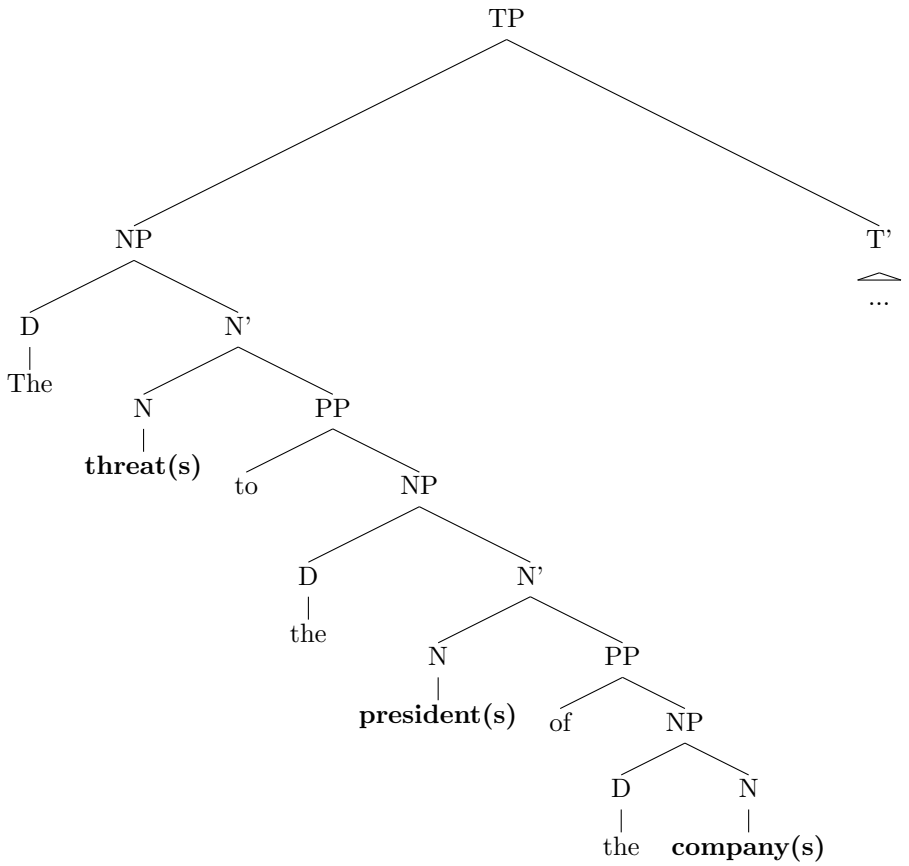


Figure 6: A simplified syntactic representation of Example 11. Even though the first attractor, the **president(s)**, is more distant from the eventual position of the verb (within the T') than the second attractor, the **company(s)**, it is closer to the verb in the syntactic structure (fewer nodes need to be crossed to reach T' from **president(s)**.

agreement attraction ($\beta = 0.23$; $|z| = 3.15$; $p < 0.005$) as well as a number asymmetry ($\beta = 0.24$; $|z| = 2.34$; $p < 0.05$), but showed no interaction between the attraction effect and the type of modifier the attractor appeared in ($\beta = 0.043$; $|z| = 0.59$; $p = 0.56$). Thus, while GPT-2's human-like overall error rates suggests that more powerful models can compute agreement more accurately overall, this increased overall accuracy does not necessarily lead to more human-like error patterns.

3.2 Syntactic vs. linear distance effects on attraction

Background: Franck et al. (2002) sought to further elucidate the role of syntactic structure in agreement attraction, focusing on a specific question: do the processes underlying agreement attraction operate over linear or hierarchical representations? To do so, they examined how attraction errors are affected by the linear distance between the attractor and verb, and compared that effect to the effect of the syntactic distance between those two words. Consider Example 11:

(11) The threat(s) [$_{PP}$ to the president(s) [$_{PP}$ of the company(s)]]. . .

This sentence contains two potential attractors: the later one, *company(s)*, appears within a PP that modifies the earlier one, *president(s)*. Since the PP that contains *company(s)* is embedded within the PP that contains *president(s)*, the path from *company(s)* to the verb along the hierarchical structure of the sentence is longer than the path from *president(s)* to that verb (see Figure 6). If we find that the lengths of these paths — what Franck et al. call the *syntactic distance* between the attractor and the verb — are inversely proportional

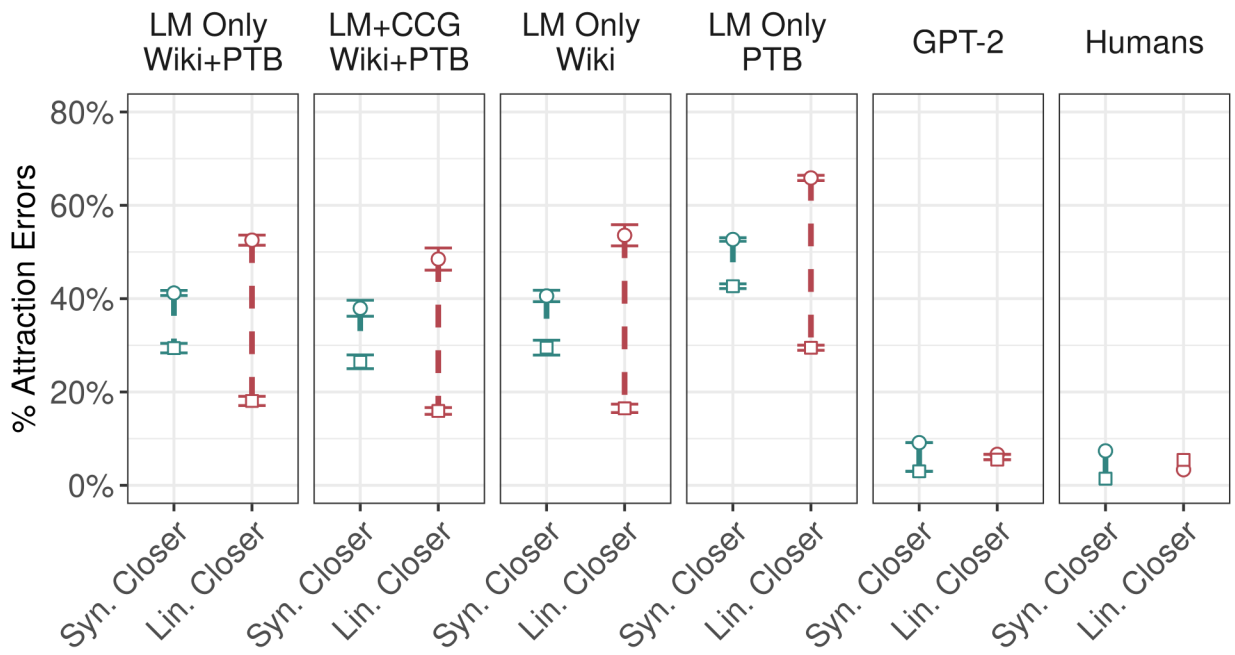


Figure 7: Human and simulation results for Franck et al. (2002). Vertical bars represent the size of the attraction effect: the difference between the subject-attractor number match condition (the lower, square endpoints) and mismatch condition (the higher, circular endpoints). These attraction effects are shown for the syntactically closer attractor (to the left of each facet) and the linearly closer attractor (to the right of each facet), marginalizing over the condition of the other attractor. Error bars represent standard errors across the five randomly initialized models trained for each model architecture and training set. Crucially, in humans, the attraction effect from syntactically closer attractors is greater than that of linearly closer attractors. The reverse is true for each of the models.

to the strength of the attraction effect caused by the two noun phrases, then we have evidence that attraction errors arise when participants process the hierarchical representations of the sentence. Franck et al. contrast these syntactic distances with the *linear distances* from the attractors to the verb, where *company(s)* is closer to the verb in the completion than *president(s)*, simply because *company(s)* appears to the right of *president(s)* in the linear sequence of words. Thus, by comparing the strength of attraction from the first, syntactically closer noun phrase (i.e., *president(s)*) to attraction from the second, linearly closer noun phrase (i.e., *company(s)*), we can investigate the nature of the structure (hierarchical or linear) used by humans or model during the agreement computations relevant to attraction: If the syntactically closer noun phrase causes stronger attraction than the linearly closer one, we have evidence for the role of hierarchical structure; if the difference is in the opposite direction, we have evidence for the role of linear order.

Human results: In Franck et al.'s experiment, syntactically closer attractors caused stronger attraction than linearly closer ones.

Simulations: The comparison of interest for each model is between the attraction effects caused by the syntactically closer attractor and that caused by the linearly closer attractor. Consequently, in Figure 7 we plot the magnitude of the attraction effect for each attractor, collapsing over the influence of the other attractor.

Both models displayed the opposite effect from humans: while there were significant effects of both the

linearly closer attractor (LM-ONLY: $\beta = 0.79$, $|z| = 38.51$, $p < 0.001$; LM+CCG: $\beta = 0.75$, $|z| = 33.57$, $p < 0.001$) and the syntactically closer one (LM-ONLY: $\beta = 0.29$, $|z| = 14.48$, $p < 0.001$; LM+CCG: $\beta = 0.28$, $|z| = 13.04$, $p < 0.001$), linear effects were significantly stronger than syntactic effects (LM-ONLY: $\chi^2 = 336.21$, $p < 0.001$; LM+CCG: $\chi^2 = 254.47$, $p < 0.001$). A comparison between LM-ONLY and LM+CCG models did not find a significant difference in either the linearly closer or syntactically closer attractor's attraction effect across LM-ONLY and LM+CCG models (linearly closer: $\beta = -0.020$, $|z| = 0.24$, $p = 0.80$; syntactically closer: $\beta = 0.013$, $|z| = 0.18$, $p = 0.86$), again indicating that, contrary to our predictions, adding the CCG training objective did not make the models' syntactic error patterns more human-like.

Effect of training corpus: Both sets of models trained on only a single corpus showed a significant effect of attraction from both the syntactically closer attractor (PTB: $\beta = 0.20$, $|z| = 8.022$, $p < 0.001$; Wiki: $\beta = 0.26$, $p < 0.001$, $|z| = 12.65$) and the linearly closer attractor (PTB: $\beta = 0.73$, $p < 0.001$, $|z| = -27.17$; Wiki: $\beta = 0.85$, $p < 0.001$, $|z| = 40.06$). However, in both cases, as in our prior experiments, the attraction effect from linearly closer attractors was much stronger than the effect from syntactically closer attractors, the reverse of what Franck et al. (2002) found in humans (PTB: $\chi^2 = 205.82$, $p < 0.001$; Wiki: $\chi^2 = 442.64$, $p < 0.001$). A comparison between the two models using two-way interactions revealed no significant differences in the attraction effect caused by either of the attractors (linearly closer: $\beta = 0.050$, $|z| = 0.53$, $p = 0.595$; syntactically closer: $\beta = -0.021$, $|z| = 0.226$, $p = 0.82$).

GPT-2: GPT-2 showed a significant effect of attraction from both the syntactically closer ($\beta = 0.41$; $|z| = 8.88$; $p < 0.001$) and linearly closer ($\beta = 0.10$; $|z| = 2.42$; $p < 0.05$). Interestingly, unlike the other models we evaluated, GPT-2 did show stronger effects from syntactically closer attractors ($\chi^2 = 24.14$; $p < 0.001$), just as human participants in Franck et al. (2002) did, as well as low overall error rates across conditions (ranging from 1.92% to 9.20%) on par with those observed in Franck et al. (2002) (approximately 1.30–9.6%).

3.3 Linear Distance Effects in Disjunction

Background: The two human experiments we have discussed so far suggested that agreement attraction in humans is sensitive to hierarchical syntactic structure, but neither provided clear-cut evidence for or against sensitivity to linear structure. In particular, in the Franck et al. (2002) comparison between linear and syntactic distance effects, syntactic distance was never held constant across linear distance conditions; as such, their results can speak only to the *relative* strengths of syntactic and linear distance, not to the existence of a linear distance effect independent of variation in syntactic distance. The absence of any linear distance effects in humans would indicate that agreement attraction errors—and, it follow, agreement computations—occur in the context of processes that operate over hierarchical structures, while the existence of a purely linear effect, over and above the hierarchical effects, would point to a hybrid set of representations that are maintained over the course of processing.

To determine if there are such purely linear effects on agreement, Haskell and Macdonald (2005) compared rates of plural agreement in sentences where the subject was a disjunction (i.e. included the word *or*), and where one disjunct was singular and the other plural (see Examples 12 and 13). Both disjuncts are equally distant from the verb in syntactic terms, under most syntactic analyses, but the second disjunct is linearly closer to the verb. As such, disjunction makes it possible to test for a linear distance effect independently of syntactic distance. Note that there is no canonical agreement pattern for disjunct subjects in Mainstream American English (see, for example, evidence from Foppolo & Staub, 2020), and thus neither the singular or plural form can be considered an agreement *error*.

(12) Can you ask Brenda if the boy or the girls…

(13) Can you ask Brenda if the boys or the girl…

Human results: Haskell and Macdonald (2005) found greater rates of plural agreement when the plural disjunct was linearly closer to the verb, indicating that linear distance affects agreement (though see Keung & Staub, 2018 for an alternative account of these results).

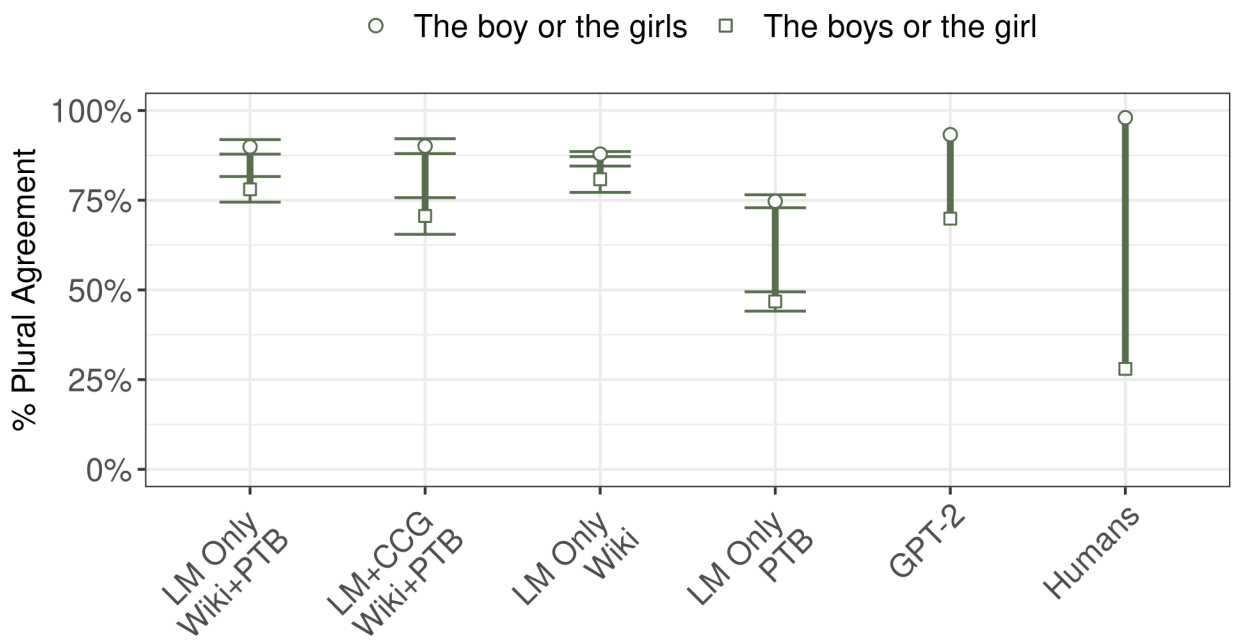


Figure 8: Human and simulation results for Haskell and Macdonald (2005). Vertical bars represent the size of the linear distance effect: the difference between plural agreement rates when the singular subject is closer to the verb position (the square endpoints) and when the plural subject is closer to the verb position (the circular endpoints). Error bars represent standard errors across the five randomly initialized models trained for each model architecture and training set. The size of the linear distance effect is represented by the length of the bar, since all models had higher rates of plural agreement when the plural subject was closer to the verb than when the singular subject was closer to the verb. While all of the models exhibited some linear distance effect, the magnitude of the effect in humans was much larger than any of the models.

Simulations: Simulation results are shown in Figure 8. Both models exhibited a similar pattern to humans: conditions where the noun closer to the verb was plural had significantly greater rates of plural agreement than conditions where the noun closer to the verb was singular (LM-ONLY: $\beta = -0.43$, $|z| = 11.22$, $p < 0.001$; LM+CCG: $\beta = -0.58$, $|z| = 12.84$, $p < 0.001$). However, the size of the effect was much smaller than that reported in Haskell and Macdonald (2005), and thus this set of results, while promising, leaves room for other models to better match human behavior. A comparison across models indicated that the CCG supertagging objective strengthened the linear distance effect compared to LM-ONLY ($\beta = 0.23$, $|z| = 4.03$, $p < 0.001$). In this case, then, the syntactic objective did lead to be more human-like behavior; surprisingly, however, this was the case for the linear distance effect rather than, as might be expected, for the hierarchical one. We return to this point in the discussion.

Effect of training corpus: Models trained on both smaller training sets also preferred to produce plural verbs when the plural disjunct appeared closer to the verb (PTB: $\beta = -0.64$, $|z| = 14.10$, $p < 0.001$; Wiki: $\beta = -0.23$, $|z| = 4.98$, $p < 0.001$).The effect size was larger in models trained on the Penn Treebank than in models trained on the much larger Wikipedia corpus ($\beta = 0.46$, $|z| = 7.59$, $p < 0.001$). This illustrates that training over larger datasets does not universally lead to more human-like behavior.

GPT-2 Performance: Like all of the other models, GPT-2 preferred producing plural verbs when the plural disjunct was closer to the verb ($\beta = -0.75$; $|z| = 8.69$; $p < 0.001$). The magnitude of this effect in GPT-2 is comparable to that found in some of the more human-like LSTM-based models (LM+CCG and LM-ONLY models trained on PTB), but is still far below that observed in humans. Since there is no canonical grammatical response in this experiment, we cannot determine whether GPT-2's sophisticated architecture leads to a reduction in error rates in this simulation.

3.4 Notional Number and Distributivity

Background: The previous experiments have characterized syntactic effects on agreement attraction: How does the linear and hierarchical position of the attractor influence agreement behavior? We now turn to semantic factors that affect agreement processing. Several studies have demonstrated an influence of *semantic* or *notional number*—the number of countable parts in the conceptual entity referred to by the noun phrase. Notional number contrasts with *grammatical number*, which is typically determined by the morphology of the head noun (e.g., the plural *-s* morpheme in many varieties of English). The role of notional number is particularly salient in collective NPs:

(14) The gang near the motorcycles...

(15) The gang on the motorcycles...

In Example 14, the preposition *near* tends to give rise to a *collective* reading, where the gang is viewed as a single collective entity located near a group of motorcycles. This gives the NP a singular notional number. By contrast, the preposition *on* in Example 15 favors a *distributive* reading, where each member of the gang is located on their own motorcycle; this results in plural notional number.

While subject-verb agreement is ostensibly a syntactic constraint, prior work has demonstrated that it is affected by the notional number of the subject, with notionally plural subjects leading to higher rates of plural agreement than notionally singular subjects (Eberhard, 1999; Bock, Nicol, & Cutting, 1999; Humphreys & Bock, 2005). Analyzing the ability of neural language models to simulate these notional number effects is of particular interest given that the models are trained solely on word prediction; since models only understand language through the text they are trained on, they lack the grounding in the physical world that might be necessary to capture agreement patterns that depend on, for instance, the spatial organization of gang members and motorcycles (Bender & Koller, 2020). Given such impoverished semantic capabilities, we hypothesize that the models will be unable to capture these semantic influences on human agreement behavior.

Human Results: In a sentence completion study, Humphreys and Bock (2005) found that participants produced plural verbs more often when the preposition favored a distributive reading (as in Example 15) than when it favored a collective reading (as in Example 14).

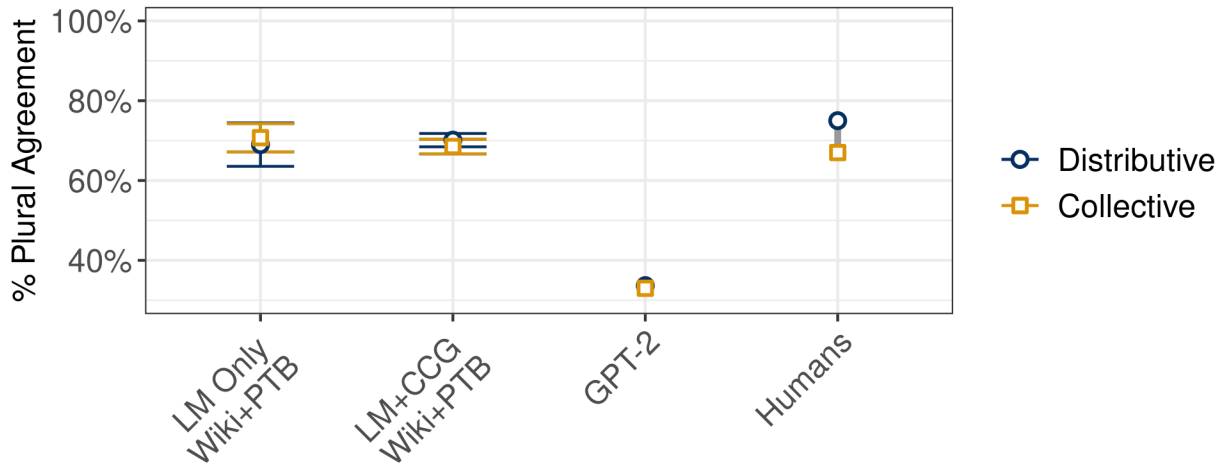


Figure 9: Human and simulation results for Humphreys and Bock (2005). Endpoints represent the rate of plural agreement in the distributive-biased condition (circular endpoints) or the collective-biased condition (square endpoints). Error bars represent standard errors across the five randomly initialized models trained for each model architecture and training set. In humans, Humphreys and Bock (2005) observed higher rates of plural agreement when the reading of the collective subject was biased toward a distributive reading. We observe no such difference in any of the models' results.

Simulation results: We compare plural agreement rates for humans and both types of models in Figure 9. Models showed no significant difference in rates of plural agreement between distributive-biased and collective-biased prepositions (LM-ONLY: $\beta = 0.047$, $|z| = 1.32$, $p = 0.19$; LM+CCG: $\beta = -0.030$, $|z| = 0.65$, $p = 0.52$), and there was no evidence of an interaction that would indicate a difference between the two types of models ($\beta = 0.074$, $|z| = 1.29$, $p = 0.20$). These null results could indicate one of two things: either our models do not use representations of notional number as part of the computations that result in an inflected verb form, or they simply have no representation of notional number at all. We will examine the second possibility in the summary of results (Section 3.7).

GPT-2 Performance: Like in our simulation of linear distance effects with disjunct subjects, there is no canonical grammatical response we should expect our models to have, so we cannot test whether the model's correctness improves. Like the other models, GPT-2 showed no differences in the rates of plural agreement between the two types of prepositions ($\beta = -0.017$; $|z| = 0.21$; $p = 0.83$).

3.5 Argument Status

Background: Agreement attraction is also affected by factors at the interface of syntax and semantics. Building on the hypothesis that *core arguments*, which are necessary for the interpretation of the verb, are encoded in memory more distinctively than *oblique arguments*, Parker and An (2018) hypothesized that the strength of attraction would differ between attractors in core arguments and attractors in oblique arguments:

(16) CORE ARGUMENT: The waitress who sat **the girl(s)** unsurprisingly was/were unhappy about all the noise.

(17) OBLIQUE ARGUMENT: The waitress who sat **near the girl(s)** unsurprisingly was/were unhappy about all the noise.

The reasoning that underlies this prediction is as follows. Memory retrieval models argue that agreement errors are caused by erroneous retrieval of the attractor's number feature of instead of that of the subject

(Badecker & Kuminiak, 2006; Wagers et al., 2009; Parker & An, 2018). Because the features that would exclude the attractor from a retrieval process targeting the subject are more likely to be more effectively encoded in core arguments, we expect the likelihood of such misretrieval to be lower when the attractor is inside a more carefully encoded core argument than when it is in less carefully encoded oblique argument. Parker and An presented participants with a sentences such as Example 16 and 17 in a self-paced reading paradigm (see Section 2.3.1). The study followed a $2\times2\times2$ design: singular vs. plural attractor, grammatical vs. ungrammatical sentence (i.e., singular vs. plural main verb), and core vs. oblique argument attractor.

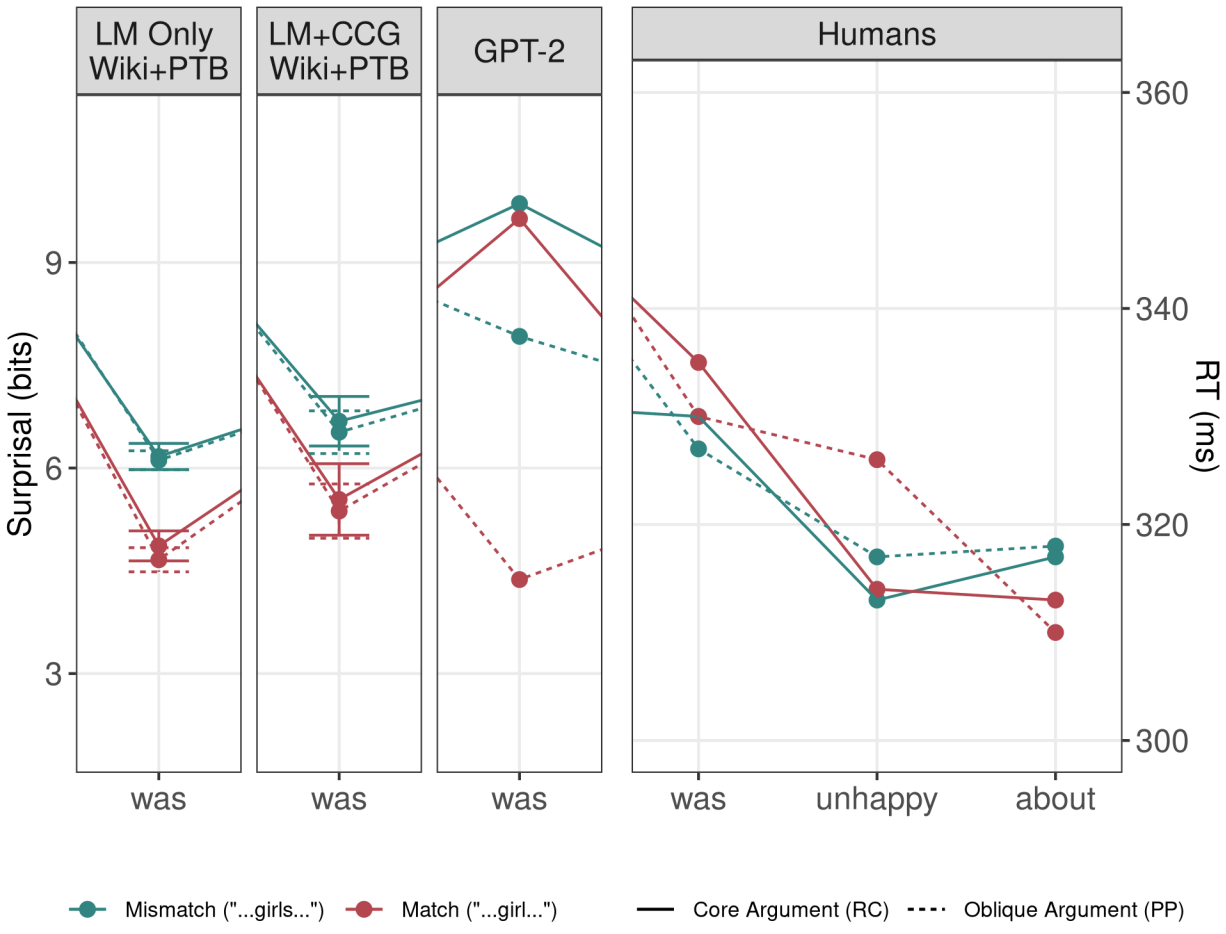
In self-paced reading, agreement attraction can manifest in two ways: first, as a facilitatory effect in ungrammatical sentences, where an ungrammatical sentence is read faster if an attractor NP is present that mismatches the subject in number (and thus matches the verb in number); and second, as an inhibitory effect in grammatical sentences, where grammatical sentences are read more slowly in the presence of an attractor NP whose number mismatches the subject (and therefore also the verb). In the ungrammatical case, the attractor creates an illusory agreeing subject-verb pair, since the attractor and verb share a number feature. Thus, in the case of an attraction error, an ungrammatical sentence is read as a grammatical one, and consequently reading times are shorter than if no error had occurred. In the grammatical case, an attraction error would result in an ungrammatical agreement relation, as the attractor and verb do not share the same number. As a result, the presence of such an error would result in longer reading times than if no error had occurred. Overall, the attractor's presence reduces the processing cost associated with ungrammaticality—the difference between reading times in grammatical and ungrammatical conditions. In the Parker and An paradigm, we expect this reduction in the cost of ungrammaticality to surface at the matrix verb (*was/were*), where the grammaticality of the agreement relation can be determined.

Human results: In Parker and An's experiment, participants were more susceptible to attraction errors when the attractors were in oblique arguments than when they were in core arguments. Parker and An do not report an analysis of reading patterns on grammatical sentences.

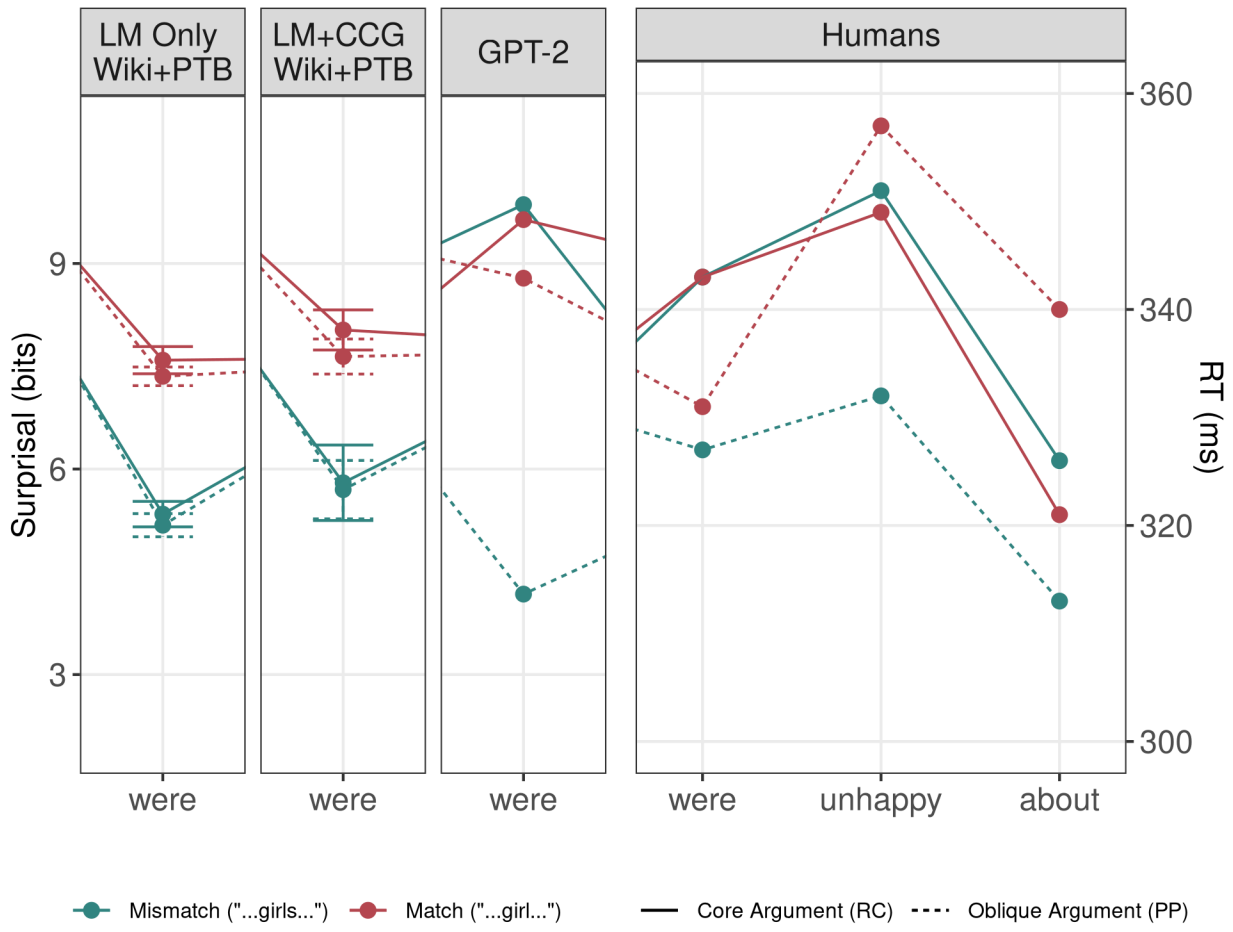
Simulation results—ungrammatical sentences: A comparison of surprisals at the critical word to the mean reading times reported by Parker and An (2018) can be found in Figure 10; for full word-by-word surprisals, and in particular the differences in surprisal at the attractor, see Appendix B.1. As in the human experiment, both models showed an attraction effect for ungrammatical oblique argument sentences (LM-Only: $\beta = -1.09$, $|t| = 26.11$, $p < 0.001$; LM+CCG: $\beta = -0.97$, $|t| = 19.17$, $p < 0.001$). Unlike humans, however, the models also showed attraction effects for ungrammatical core argument sentences (LM-Only: $\beta = -1.12$, $|t| = 27.80$, $p < 0.001$; LM+CCG: $\beta = -1.12$, $|t| = 22.19$, $p < 0.001$), and there was no significant interaction between argument status and attraction (LM-Only: $\beta = -0.018$, $|t| = 0.615$, $p = 0.53$; LM+CCG: $\beta = -0.072$, $|t| = 1.94$, $p = 0.051$). An analysis comparing LM-only and LM+CCG models did not find a significant three-way interaction between model type, argument type and number mismatch ($\beta = 0.053$, $|t| = 1.12$, $p = 0.26$), suggesting that the syntactic training objective did not affect the models' ability to simulate the human error patterns.

Simulation results—grammatical sentences: As Parker and An do not present attraction analyses for the grammatical sentences in their experiment, we present the simulation results here without comparing them to the human patterns. Both models showed a significant effect of attraction (LM-Only: $\beta = 0.69$, $|t| = 24.00$, $p < 0.001$; LM+CCG: $\beta = 0.57$, $|t| = 15.62$, $p < 0.001$), but no significant interaction between attraction and argument status (LM-Only: $\beta = -0.037$, $|t| = 1.28$, $p = 0.20$; LM+CCG: $\beta = -0.0024$, $|t| = 0.064$, $p = 0.95$). A comparison between LM-Only and LM+CCG did not find a three-way interaction between the additional objective, attractor argument type, and subject-attractor number match ($\beta = -0.034$, $|t| = 0.73$, $p = 0.46$). It did, however, yield an interaction between the model type and subject-attractor number match, reflecting smaller attraction effects in LM+CCG ($\beta = -.0012$, $|t| = 2.15$, $p < 0.05$).

GPT-2: For this (and the following) comprehension simulation, there is no real measure of a model's error rate. As a result, these results cannot show whether GPT-2 has a lower overall error rate relative to our LSTM models. We thus present results of these simulations only to demonstrate the ability of GPT-2 to mimic human error patterns.



(a) Simulation Results, Grammatical



(b) Simulation Results, Ungrammatical

Figure 10: Word-by-word surprisals from our simulations and corresponding reading times from Exp. 1 of Parker and An (2018). Error bars are standard errors. Since effects in self-paced reading typically spillover into the reading times of the next few words, we provide two additional words for the human results. The relevant effect is found at the *unhappy* in the human data, with the attraction effect in the oblique argument condition (the difference between dashed lines) being significantly larger than the attraction effect in the core argument condition (the difference between solid lines). We see no such difference in the models.

In ungrammatical sentences, we found a significant attraction effect ($\beta = -1.10$; $|t| = 7.01$; $p < 0.001$), with an interaction with argument status such that the attraction effect was attenuated when the attractor was in core arguments compared to oblique arguments ($\beta = 1.21$; $|t| = 7.71$; $p < 0.001$). Grammatical sentences displayed a similar pattern, with a significant attraction effect ($\beta = 0.94$; $|t| = 5.70$; $p < -0.001$) that was smaller when the attractor was in a core argument ($\beta = -0.83$; $|t| = 5.039$; $p < 0.001$). Unlike the other models, then—and like humans—GPT-2 showed an effect of argument status on attraction effects, suggesting that GPT-2's larger training set, transformer architecture, or some combination of these factors allows GPT-2 to represent argument status, as well as encode that feature in a way that influences agreement processing.

3.6 Grammaticality Asymmetry

Background: As noted in the previous section, attraction can affect reading in two ways: it can cause participants to read grammatical sentences more slowly, or it can cause them to read ungrammatical sentences faster. Theories that attribute agreement attraction to an error in encoding the number of the subject (Eberhard et al., 2005, among others) predict that both of these effects should be of the same magnitude (Badecker & Kuminiak, 2006; Wagers et al., 2009). This is because grammaticality is determined by the number of the verb, which appears only after the subject is encoded; consequently, we expect an equal number of subject encoding errors to occur in grammatical and ungrammatical sentences.

Some encoding accounts also hypothesize that encoding errors emerge from an erroneous percolation of the attractor's number feature to the subject noun phrase as a whole (Franck et al., 2002). These accounts thus additionally predict that attraction errors can only occur when the attractor is within the subject NP, as that is the only way there is an upward path through which the attractor's number feature can percolate to the subject's node in the sentence's syntactic structure.

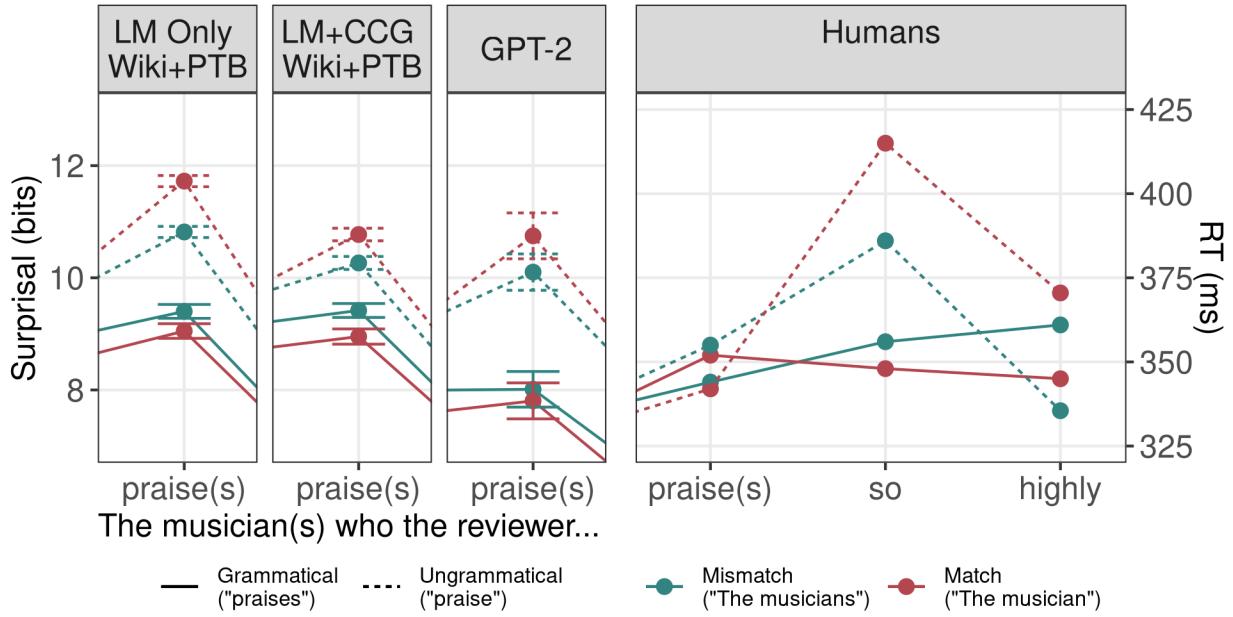
Wagers et al.'s self-paced reading study tests both of these predictions using sentences with RC-modified subjects:

(18) The musician(s) [who the reviewer(s) praise(s) so highly] will probably win a Grammy.

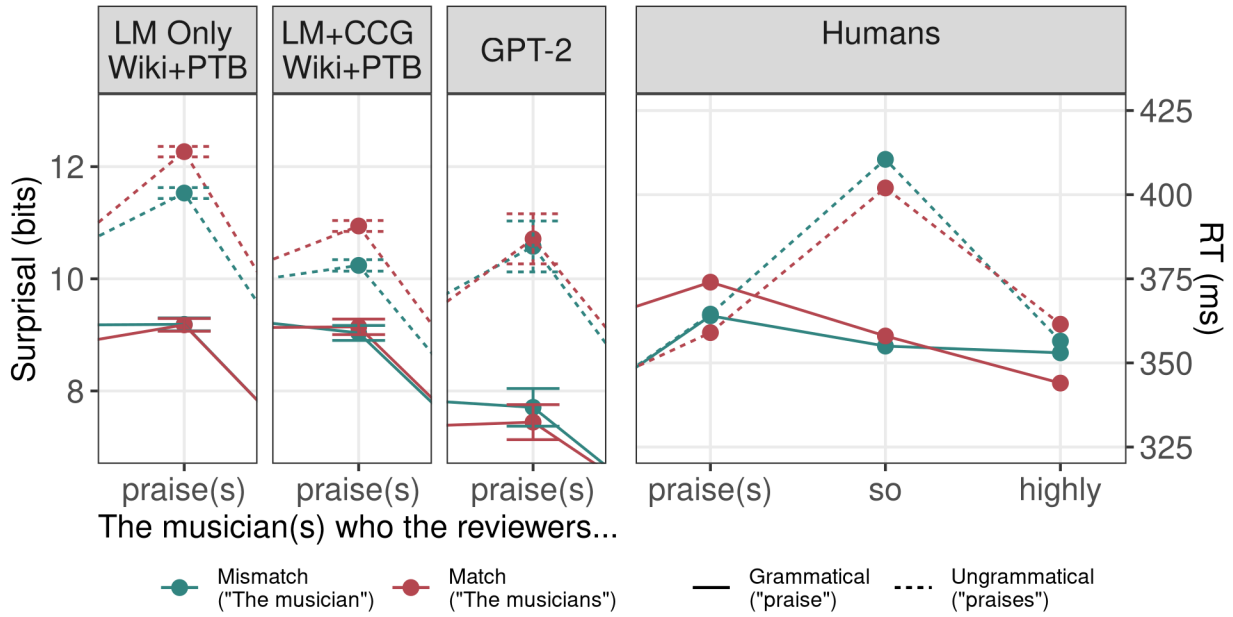
Unlike the sentences used in the Bock and Cutting (1992) experiment discussed above, in these materials the matrix clause subject, *musician(s)*, acts as the attractor NP, and the agreement relation that is manipulated—the subject-verb dependency between *reviewer(s)* and *praise(s)*—is internal to the relative clause. As a result of this configuration, the attractor is not within the subject, and thus percolation accounts predict no attraction in this paradigm.

Human results: Contrary to the predictions of the encoding account of agreement attraction, Wagers et al. (2009) found that human readers show a *grammaticality asymmetry*: they displayed attraction effects in ungrammatical sentences, but not in grammatical ones. Wagers et al. (2009) additionally confirmed that attractors outside of a relative clause can cause attraction within that relative clause, providing additional evidence against percolation accounts.

Simulation results: A comparison between the models' surprisals at the critical word and reading times at the critical region of the human data can be seen in Figure 11. Full word-by-word surprisals, which show surprisal differences due to words prior to the critical region, can be seen in Appendix B.2. Like humans, both types of models showed a significant agreement attraction effect in ungrammatical sentences (LM-Only: $\beta = -0.41$, $|t| = 12.48$, $p < 0.001$; LM+CCG: $\beta = -0.30$, $|t| = 10.17$, $p < 0.001$), but, unlike the humans, they also showed attraction in grammatical sentences (LM-Only: $\beta = 0.09$, $|t| = 3.32$, $p < 0.005$; LM+CCG: $\beta = 0.089$, $|t| = 3.02$, $p < 0.005$). We found a significant interaction between attraction and grammaticality in both models (LM-Only: $\beta = -0.16$, $|t| = 6.72$, $p < 0.001$, LM+CCG: $\beta = 0.107$, $|t| = 4.83$, $p < 0.001$), where ungrammatical sentences displayed larger attraction effects than grammatical sentences, in line with the grammaticality asymmetry observed in humans. An analysis comparing the simulation results across types of models found no evidence of an effect of CCG supertagging objective on the grammaticality asymmetry ($\beta = -0.054$, $|t| = 1.57$, $p = 0.11$). The presence of an asymmetry indicates that, like humans, agreement errors in models are not simply caused by faulty encoding of the subject's



(a) Simulation Results, Singular Subject



(b) Simulation Results, Plural Subject

Figure 11: Surprisals for models in our simulation of Exp. 3 of Wagers et al. (2009) at the verb *praise(s)*, where the grammaticality of the agreement relation within the RC becomes clear. Error bars are standard errors. We see a grammaticality asymmetry in both humans and models, as the attraction effect in ungrammatical sentences (the difference between dashed lines) is greater than that in grammatical sentences (the difference between solid lines).

number, but by a mechanism that is sensitive to the verb's number. This could take the form of a retrieval error, as Wagers et al. argue is the case for humans, or a bias toward reading sentences as grammatical (Hammerly, Staub, & Dillon, 2019). We return to this point in Section 3.7.1.

GPT-2: Unlike the rest of the models we evaluated, GPT-2 failed to find a significant attraction effect in either ungrammatical sentences ($\beta = 0.39$; $|t| = 1.46$; $p = 0.15$) or grammatical sentences ($\beta = -0.23$; $|t| = 1.18$; $p = 0.24$), and there was no significant interaction between attraction and grammaticality ($\beta = -0.16$; $|t| = 0.44$; $p = 0.66$). We did find a significant attraction effect when considering only sentences with a singular subject, and thus a plural attractor in the mismatch condition ($\beta = 0.65$; $|t| = 2.33$; $p < 0.05$), the condition where we would expect the largest attraction effects due to a combination of number asymmetry and grammaticality asymmetry effects.[5]

3.7 Summary of Results

The simulations we reported in this section aimed to answer three major questions: first, what phenomena from the human agreement attraction literature are captured by a simple, syntactically un-biased neural network language model (LM-Only)? Second, does the addition of an explicitly syntactic training objective lead models to better capture those phenomena? And third, how do differences in the corpora used to train a neural language model affect the phenomena the model captures? In this section, we discuss how the results of our six simulations bear on these three questions. We then contextualize our findings more broadly in the General Discussion.

3.7.1 What phenomena do LM-Only models capture?

Our first goal was to determine how well a simple language model that lacks explicit language-specific biases captures the range of factors that affect agreement processing in humans. To do so, we compared the behavior of human participants to the behavior of LM-Only models trained on both Wikipedia and the Penn Treebank. The experiments we simulated can be grouped into three categories: experiments that bear on the role of hierarchical structure in agreement processing, experiments that bear on the role of semantic factors in agreement processing, and an experiment that demonstrates a grammaticality asymmetry in agreement attraction. For clarity of presentation, we will start with a discussion of the grammaticality asymmetry and conclude with a discussion of experiments testing the role of hierarchical structure; we discuss the effect of additional syntactic training in the next section.

The Grammaticality Asymmetry In our simulation of Experiment 3 from Wagers et al. (2009), we sought to determine whether models can simulate the grammaticality asymmetry, where attractors cause ungrammatical sentences to be read faster but do not cause grammatical sentences to be read more slowly. We found that models—both LM-Only and LM+CCG—behave in line with this asymmetry, displaying greater susceptibility to attraction in ungrammatical than grammatical sentences.

Wagers et al. interpret the grammaticality asymmetry in humans as indicating that attraction does not result solely from encoding errors. In English, subjects generally precede the verbs they agree with. As a result, an error in encoding the subject's number necessarily occurs before the verb is processed, and therefore the number of the verb (and thus the grammaticality of the subject-verb agreement relation) should make no difference in the rate of agreement errors: we should see as many errors in grammatical sentences as in ungrammatical ones. The fact that we do see a grammaticality asymmetry, Wagers et al. argue, lends credence to alternative models of agreement attraction, such as those that posit that errors emerge during the retrieval of the subject's number.

Wagers and colleagues' account of the grammaticality asymmetry could plausibly explain the models' behavior. Our language models can be divided into two components: an LSTM encoder, which constructs a representation of the sequence of words observed thus far, and a decoder, which takes the representation generated by the encoder and outputs a probability distribution over the next word. The distinction between these two components roughly corresponds to the distinction between encoding and retrieval processes: the LSTM encoder, like encoding processes in human participants, only has access to the subject when

[5]This is a replication of one of the simulations reported by Ryu and Lewis (2021).

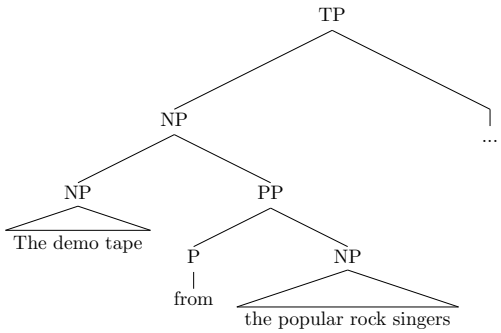
constructing its encoding, while the decoder's estimate of a verb's likelihood as the next word is sensitive to the identity of the verb (i.e., our models' estimate of $P(w^*_{i+1} \mid w_1, \ldots, w_i)$ is sensitive to the hypothetical next word w^*_{i+1}). Since this probability is directly mapped to our simulated behavioral measure (as described in Sections 2.3.2 and 2.3.1), we can use Wagers and colleagues' reasoning to conclude that some of the erroneous behavior of the models must be attributed to the decoder rather than the encoder, as the asymmetry is only possible if the process generating the errors can determine the number (and thus the grammaticality) of the verb.

Factors at the Syntax-Semantics Interface We simulated two human experiments that were concerned with factors at the syntax-semantics interface: distributivity in agreement with collective subjects (Humphreys & Bock, 2005) and the effect of argument structure on agreement attraction (Parker & An, 2018). Our models both failed to mirror human behavior in both simulations: there was no difference in plural agreement rates between distributive-biased and collective-biased subjects, and no difference in attraction rates between attractors in core and oblique arguments. We hypothesize that the effects on models' failure to simulate these semantic effects on agreement is connected to a more fundamental issue in language models: The inability of models trained solely on language modeling to develop the grounding necessary for true language understanding (Bender & Koller, 2020). In particular, to match the hypothesized mechanism underlying human behavior for the distributivity experiments (Humphreys & Bock, 2005), a model would need to distinguish between, for example, an NP that is more likely to be conceptualized as a single, collective entity and an NP that is more likely to be conceptualized as multiple entities distributed in space. This kind of mapping, from linguistic material to entities in an external world, may lie beyond the abilities of models trained solely on linguistic material. We speculate that a multi-modal model with a visual training objective may be better able to capture such effects (for a example of a multi-modal model in distributional semantics, see Bruni, Tran, & Baroni, 2014).

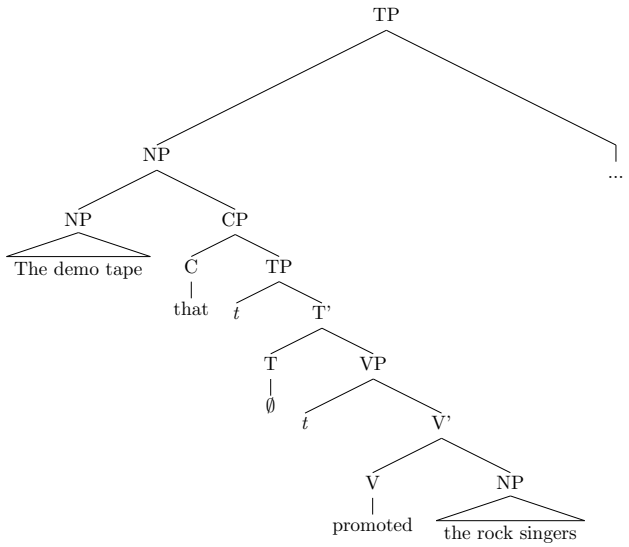
Similar limitations could be assumed to underlie the models' failure to simulate the results of Parker and An (2018). The difference between attractors in core and oblique arguments in humans is hypothesized to be due to the differential encoding of arguments based on their importance during interpretation: since core arguments are more central to interpretation than oblique ones, attractors in core arguments are better encoded, and thus are less likely to interfere with agreement than more poorly encoded oblique arguments. Since word prediction models are never tasked with interpreting the representations they construct, they face no pressure to differentially encode core and oblique arguments, which may explain why this distinction does not affect the models' agreement error rates. However, this explanation is complicated by our simulations using GPT-2, which reveal differences in attraction from core and oblique arguments. We leave an exploration of exactly how this behavior manifests in GPT-2 to future work.

Hierarchical Structure and Linear Distance The first three experiments we simulated characterized the effect of syntactic and linear position on agreement attraction: differences in attraction strength between attractors in prepositional phrases and relative clauses (Bock & Cutting, 1992), differences in syntactic distance between the attractor and verb (Franck et al., 2002), and differences in linear distance between disjuncts in the subject and the verb (Haskell & Macdonald, 2005). LM-Only models broadly failed to capture these structural effects: Our simulations found no difference between attraction effects for PP attractors and RC attractors, whereas humans made more attraction errors for preambles with PP attractors compared to those with RC attractors (Bock & Cutting, 1992). Our simulations also showed stronger attraction effects from attractors linearly closer to the verb than ones that were syntactically closer to the verb—the reverse of the effect found by Franck et al. (2002). Taken together, these two results suggest that models operate over linear representations based on the surface form of the input rather than the hierarchical representations used by humans (Momma & Ferreira, 2019). Finally, though the models displayed a significant effect of linear distance in the same direction as the effect found by Haskell and Macdonald (2005), the magnitude of this effect was far smaller than in humans.

We hypothesize that stronger hierarchical biases may be necessary for models to fully simulate syntactic and linear distance effects on human agreement processing. The two empirical findings we failed to capture—the effect of the type of modifier in which the attractor appears (PP vs. RC), and the effect of the depth of the attractor within the subject—can both be explained through syntactic distance (Franck et al., 2002), under the assumption that higher rates of agreement errors correspond to a shorter distance from the attractor



(a) Example PP-modified Preamble



(b) Example RC-modified Preamble

Figure 12: Example (simplified) syntactic trees corresponding to the PP and RC conditions in Bock and Cutting (1992). Crucially, the attractor NP in embedded more deeply in the subject's structure in the RC-modifier condition (12b) than in the PP-modifier condition (12a), resulting in a longer syntactic distance from the attractor to the inflected verb's position.

to the verb in the hierarchical structure of the sentence (see Figure 12). This suggests that what may be missing from our models is an accurate hierarchical representation of input with a strong causal role in the models' word predictions: if the models compute agreement over a flat, linear representation, they cannot be sensitive to differences in a measure such as syntactic distance. LM+CCG models, which were trained with explicit syntactic supervision (CCG supertagging), were motivated by this hypothesis; we discuss those models in the next section.

3.7.2 Does the syntactic bias imparted by supertagging lead to more human-like behavior?

Success at the supertagging task requires sophisticated representations of syntactic structure. For example, correctly predicting the supertag (S\NP)/ADJ for "is" in "The key to the cabinets is..." requires a model to both recognize an NP on the right periphery of the left context (either "the cabinets" or "the key to the cabinets") and predict that the upcoming material will eventually result in an ADJ (i.e., the next token being "rusty") that combines with the current word and the NP to the left to form an S (i.e., the sentence "The key to the cabinet is rusty"). We hypothesized that a language model that shared the representations it uses for word prediction with a supertagger would be biased toward accessing the syntactic information in those representations, and, as a result, would exhibit more human-like error patterns when simulating agreement attraction experiments, particularly those that tested syntactic phenomena (Bock & Cutting, 1992; Franck et al., 2002). This hypothesis was not borne out: the syntactic training objective had no discernible impact on the ability of the models to capture human error patterns in our simulations of Bock and Cutting (1992) and Franck et al. (2002). At the same time, this objective did lead to more human-like results in other simulations: LM+CCG models exhibited a stronger number asymmetry (Bock & Cutting, 1992), stronger linear distance effects (Haskell & Macdonald, 2005), and weaker attraction in grammatical sentences (Parker & An, 2018) than LM-ONLY models. We discuss each of these observations in turn.

Are representations shared between word prediction and supertagging? Why did the supertagging objective fail to affect the networks' syntactic behavior? Our hypothesis was that in the multi-task setting the representations generated by the LSTM encoder would better encode fine-grained syntactic information; those, in turn, would be used not only by the classifier that performed the supertagging task, but

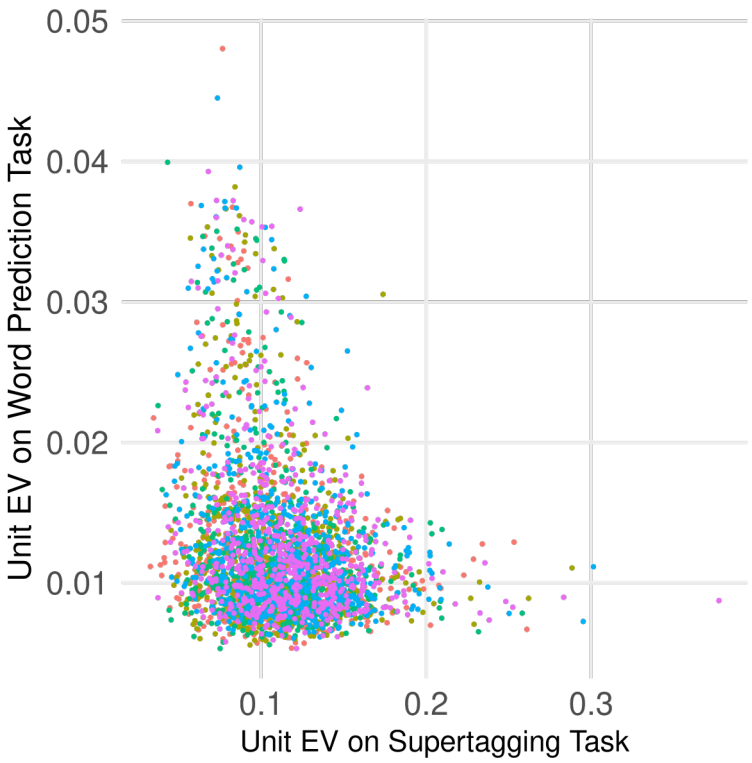


Figure 13: A scatterplot of each unit's variance across words in the word prediction objective's linear classifier (the Unit *Efferent Variance* on word prediction) against that unit's variance across supertags in the CCG supertagging objective's linear classifier (the Unit Efferent Variance on supertagging). Colors indicate the individual model instance a unit belongs to.

also by the classifier dedicated to word prediction, which determines the overall behavior of the cognitive model. This hypothesis crucially rests on the assumption that the representations used by the two classifiers are shared; if that assumption is incorrect, and the two sets of representations are distinct, separable subspaces of the LSTM encoder's representational space, we would expect little difference in the syntactic behavior of LM-Only and LM+CCG models during word prediction.

To test whether the limited impact of the supertagging objective was due to the lack of shared representations between the two objectives, we examined the efferent (outgoing) connections between each of the units in the final layer of the LM+CCG models' LSTM encoder and its two decoders: the classifier predicting the next word and the classifier determining the supertag of the current word. Each of these connections—between an LSTM unit and a specific word or CCG supertag—is associated with a learned weight. We hypothesized that LSTM units with higher variance across the efferent connection weights from the unit to the word prediction layer would be better able to differentiate between upcoming words, and consequently the classifier would be likely to use the information encoded in that unit to predict the next word. A similar logic applies to the variance of the efferent weights leading to the supertag classifier. We refer to this quantity as the unit's *efferent variance*, or EV, with respect to the model's word or supertag prediction. If representations are shared across the two objectives, we expect to find a gradient correlation between the LSTM units' EV for the two tasks (the higher the suppertagging EV, the higher the word prediction EV); alternatively, we might expect to find a subset of syntactic units that influence both word prediction and supertagging tasks, and as such have high EV across both tasks.

The results of this analysis were negative: We did not find evidence of a positive correlation between the units' EV on the two tasks, nor a subset of units with high EV across both tasks. Within each of the five LM+CCG models, correlations between the units' word prediction EV and their supertagging EV ranged from -0.212 to -0.288, indicating that a unit's EV for word prediction is, if anything, inversely correlated

with its EV for supertagging. In addition, plotting each unit's word prediction EV against its supertagging EV (Figure 13) showed no evidence of a subset of syntactically aware units that might be driving predictions for both tasks. As a result, we find it plausible that the syntactic information used to predict supertags is not used for word prediction, which may explain the lack of difference between the simulations of syntactic experiments with LM-Only and LM+CCG models. We will discuss the potential implications of this lack of shared representations, along with potential methods to encourage greater sharing of representations, in Section 4.1.

Simulations where LM+CCG models were more similar to humans While we found little difference between LM-only and LM+CCG models in the simulations that bear on linear and syntactic distance, we did find three notable differences between the models' performance, all of which bring LM+CCG models closer to human-like error patterns.

First, in our simulation of Bock and Cutting (1992), LM+CCG models exhibited a larger number asymmetry than LM-only models—like humans, the models showed larger attraction effects when attractors were plural than when they were singular. Second, in our simulation of Haskell and Macdonald (2005), while LM-only models showed a human-like effect—the number of the linearly closer attractor in a disjunct subject like *the boys and the girls* biased the speaker to produce a verb with the same number—the magnitude of this effect was much smaller than that in human participants. LM+CCG models showed a larger effect size for this experiment, though it was still not comparable to that of humans. Finally, in our simulation of Parker and An (2018), LM+CCG models showed smaller agreement attraction effects in grammatical sentences than LM-only models, while the attraction effect in ungrammatical sentences did not change significantly. While Parker and An (2018) do not report human results for grammatical sentences, this pattern is in line with the grammaticality asymmetry observed in Wagers et al. (2009), where agreement attraction was found only in ungrammatical sentences.

Overall, while contrary to our expectations the differences between LM-only and LM+CCG models did not emerge in the simulations that focused on syntax, we find it important to note that LM+CCG models were either no different from LM-only models or were more similar to humans than LM-only models. We return to this point in Section 4.1.

To understand these differences in light of our EV analysis, it is helpful to consider the various ways in which an additional supertagging objective can influence our model's word prediction behavior. We hypothesized that the additional supertagging task would give the model additional incentive to learn syntactic representations for supertagging that will then be recruited for word prediction. Our EV analysis in the previous section suggests that this has not happened, since the supertagging and word prediction parts of our model appear not to share any information

Alternatively, the supertagging task may only indirectly affect word prediction behavior by competing with the word prediction parts of the model during training. What representations the model can learn for word prediction are constrained by our model's need to learn representations for both word prediction and supertagging, regardless of whether those two representations overlap. These constraints can manifest in a variety of ways, including competition over the fixed number of units that must be split or shared amongst the two tasks. We leave it to future work to investigate how these constraints can lead to different (and potentially more human-like) patterns of errors in word prediction.

3.7.3 How does training data affect agreement behavior?

The behavior of a neural network is determined not only by its architecture and training objective, but also by its training setup, and in particular the data used to train it. This section discusses the results of our experiments that compared LM-Only models trained on the Wall Street Journal section of the Penn Treebank to those trained on a subset of English Wikipedia. These two training corpora differ in both size and genre; we will discuss these factors in turn.

The first difference between the corpora is dataset size. Whereas the Wall Street Journal section of the Penn Treebank is composed of just under 1 million words, the subset of English Wikipedia is significantly larger, consisting of approximately 80 million words. In general, models that are given more data learn to perform better at their task (Kaplan et al., 2020), and models that perform better at their task tend to behave in a more human-like manner (Merkx & Frank, 2021). We see this in models trained on the Wikipedia

dataset, which show more human-like agreement behavior than models train on the Penn Treebank in our simulation of Bock and Cutting (1992).

In order to investigate the effect of dataset genre, we estimated the frequency of a number of relevant agreement configurations (subject-verb relations, relative clauses, disjunct subjects, etc.) for each of our simulations within the Penn Treebank as well as a subset of 500,000 sentences from the Gulordava et al. (2018) Wikipedia corpus. We do this under the assumption that the training dataset influences the final model's agreement behavior primarily by exposing the model to various agreement-related syntactic configurations. We presume that greater exposure to these configurations will lead to more human-like behavior for simulations that rely on properties of those configurations (i.e., models will better learn the syntactic structures associated with relative clauses if they see more relative clauses during training). We parsed each sample of sentences from each corpus using the Chen and Manning (2014) dependency parser, and checked each resulting parses for each of the relevant syntactic configurations. The resulting counts are displayed in Table 1. Note that, since the counts were derived from the output of an automatic parser, which may contain errors, they serve only as approximate estimates of the relevant frequencies.

One of the largest differences in structural frequency between the two corpora is in the case of disjunct subjects. Contrary to our expectations, we see a higher frequency of disjunct subjects in the Wikipedia corpus than in the Penn Treebank, suggesting that the Penn Treebank models' human-like performance in our simulation of Haskell and Macdonald (2005) is not due to more extensive exposure to this construction. Instead, it appears that greater exposure, and thus presumably more sophisticated representations of disjunct subjects, leads to more consistent plural verb responses regardless of the plural disjunct's position (this contrasts with humans, who are biased towards the number of the closer disjunct). This behavior appears to be consistent with traditional structural accounts of coordination where both disjuncts are assumed to be in a symmetric relationship, and as such linear position is irrelevant for operations like agreement (e.g., Williams, 1978). By contrast, a weaker representation of disjunction leads to more uncertainty as to the number of the verb the model chooses to predict, leading to more mixed predictions when singular disjuncts are closer to the verb.

Its important to note that what we aim to show here is not a particular mechanism for how structural frequency in a training corpus leads to particular processing behaviors, but that differences in the size and syntactic distribution of the dataset we use to train a model influence the model's syntactic processing behavior. In these two datasets, we see fairly small differences in the distribution of many of the structures under consideration; the other notable difference in structural frequency across datasets concerns RCs, the other case in which the Wikipedia-trained and PTB-trained models differ in behavior (in the simulation of the PP/RC asymmetry). If we aim to replicate the learning conditions of humans, however, we must acknowledge that the style of Wikipedia and the Wall Street Journal (i.e., formal and edited written text) is likely far different in distribution from what is typical of spoken language or child-directed speech. This emphasizes the importance of considering the data used to train models – both in size and in composition — when evaluating how those models learn to process language. Alternatively, one can aim to design models with inductive biases strong enough to develop human-like processing characteristics with some robustness as to the training data provided. We will return to this point in the General Discussion.

3.7.4 What improvements does GPT-2 model show relative to our LSTM-based models?

We compared our LSTM-based models (LM-Only and LM+CCG) to GPT-2, a much larger and more powerful language model. GPT-2 differs from our models in multiple ways: the number of training samples, the number of learned weights, and the models' architectures. As such, it is difficult to draw conclusions about the sources of the differences in behavior between the GPT-2 and each of our models. We can, however, use GPT-2 to address two related questions: first, is the LSTM models' high rate of agreement errors specific to these models or is it a property of neural networks more broadly? While in the present work we prioritized an investigation of the *qualitative patterns* of errors, a long-term goal of this research program is arguably to also provide a quantitative match to human error patterns; if neural networks' overall agreement error rates are uniformly much higher than those of humans, this goal is unlikely to be met. If GPT-2's overall error rates are indeed lower, we can also address a second question: Is there a relationship between overall error rates and the qualitative match between model and human error patterns?

In the PP vs. RC experiment of Bock and Cutting (1992) and the syntactic distance experiment of

	PTB		Wikipedia	
	count	per sentence	count	per sentence
Sentences	42068	1	500000	1
Subject-Verb relations	64694	1.54	658173	1.32
Number-marked agreement relations	17421	0.41	134362	0.27
RC subject modifiers	1427	0.034	8963	0.018
PP subject modifiers	7519	0.18	76708	0.15
Nested PP subject modifiers	1027	0.024	10091	0.020
Disjunct subjects	96	0.0023	1746	0.0035

Table 1: Counts of relevant syntactic phenomena in the Penn Treebank and a subset of Wikipedia. Number-marked agreement relations are those in which a clear number feature is tagged by the parser for both the head of the subject and verb, and thus can teach the models about agreement. This is not the case in, for instance, the English past tense, where verbs are not marked for number (*the dogs barked* and *the dog barked* are both grammatical).

Franck et al. (2002), GPT-2 did in fact exhibit overall error rates comparable to humans. This indicates that the failure of our models to reach comparable overall error rates is due not to a fundamental issue with neural network models broadly, and reinforces the promise of a careful, incremental study of the inductive biases that lead to human-like error patterns in neural nets. GPT-2's higher overall agreement accuracy does not, however, allow us to ignore accuracy when analyzing error patterns. Better performance is deeply tied to having sophisticated syntactic representations, since those representations help to identify verbs and their corresponding subjects. Thus when we manipulate a model's inductive bias to improve its syntactic representations, we make it easier for the models to have a higher overall accuracy on agreement tasks. Similarly, models with weaker syntactic inductive biases are likely to have impaired performance on agreement because of their weak syntactic representations in addition to whatever error patterns they exhibit.

This leads us to our second question: If better overall agreement accuracy is tied to inductive biases that lead to human-like error patterns, do more powerful models like GPT-2 always have more human-like error patterns? The answer to this question is no. In our simulations of Bock and Cutting (1992), Haskell and Macdonald (2005) and Humphreys and Bock (2005), GPT-2's errors did not match the human error pattern any more than the LSTM-based models did; worse, in our simulation of Wagers et al. (2009), GPT-2 failed to show the grammaticality asymmetry we found in all of our LSTM-based models. At the same time, the error patterns in the remaining two experiments did match the human one more closely. In our simulation of Franck et al. (2002), GPT-2 showed greater attraction effects from syntactically closer attractors than linearly closer ones. In our simulation of Parker and An (2018), attraction effects were greatly attenuated when attractors appeared in core arguments compared to oblique ones. We see these differences as worthy of further investigation, particularly in light of accounts comparing the mechanisms of transformer-based models such as GPT-2 and cue-based models of memory retrieval (Ryu & Lewis, 2021).

Overall, the dissociation we find between the models' syntactic overall competence, their language modeling performance (accuracy of word predictions), and their match to human behavioral patterns is convergent with prior work indicating that language modeling ability does not predict scores on syntactic benchmarks (Hu et al., 2020) and that performance on those syntactic benchmarks does not correlate with models' ability to predict human behavioral measures like reading times or eye-movements (E. G. Wilcox, Gauthier, Hu, Qian, & Levy, 2020).

4 General Discussion

In this paper we have proposed a framework for employing neural networks as broad-coverage models of human syntactic processing, and have used this framework to compare the errors made by humans in a suite of studies from the English subject-verb agreement processing literature to the errors made by two classes of networks: first, LM-ONLY models, which were trained solely on word prediction over a text corpus; and

second, LM+CCG models, which were trained on word prediction as well as CCG supertagging, a task that requires sophisticated representations of syntactic relationships between words, and thus, we reasoned, should impart a syntactic inductive bias to model's word prediction component.

Both classes of models successfully simulated some human results, but failed to simulate others. They were especially unsuccessful in replicating human error patterns that can be attributed to syntactic structure; contrary to our hypothesis, LM+CCG models did not show more sophisticated, human-like syntactic performance than LM-ONLY models, although they did perform in a more human-like manner than LM-ONLY models in some of the simulations that were not directly linked to syntactic structure. Follow-up analyses indicated that the network's internal representations may not be shared across the word prediction and CCG supertagging components, suggesting, as we discuss in detail below, that stronger integration between the syntactic and word prediction component of the model will be necessary to impart a syntactic bias to the word prediction component.

Neural networks' representations depend not only on their inductive biases (i.e., their architecture or training objectives), but also on their training data. To assess the sensitivity of our results to the training corpus, we repeated a subset of our simulations using language models trained only on 80 million words of English Wikipedia, or only on the approximately one million words of the Penn Treebank. While these datasets differed both in size and style, the architectures of the models trained on those datasets were held constant; as such, differences in the models' agreement error patterns could indicate that the models' inductive biases are too weak to properly constrain the syntactic behavior the models learn from the kinds of naturalistic text we use as training data. Our analyses confirm this concern: Models trained only on Wikipedia did not consistently exhibit more or less human-like syntactic behavior than models trained only on the smaller Penn Treebank subset, indicating that the behaviors our models learn are sensitive to both training set size and style.

In the sections below, we will discuss these findings and their implications more broadly. We will then consider the potential for the use of neural network language models as cognitive models of syntactic constraints like agreement, as well as the possible pitfalls and best practices that emerge from our experiments.

4.1 Does adding syntactic inductive biases lead to more human-like syntactic performance?

To test whether additional syntactic biases would lead models toward more human-like syntactic processing, we trained language models with a second, explicitly syntactic training objective: CCG supertagging. This choice was predicated on the hypothesis that this second objective would create pressure on the model's internal representations to encode the syntactic relationships between the words of the sentence in a more accurate fashion. To the extent that those internal representations are shared between the word prediction and supertagging tasks, we hypothesized that those more syntactically detailed internal representations would lead to more syntactically informed behavior in the word prediction task as well, in turn leading to more human-like performance. Crucially, since in our framework error rates are based only on the output of the word prediction component, for the auxiliary task to affect agreement error patterns, LM+CCG models must use the syntactic information encoded in their internal representations not only for CCG supertagging but also for word prediction. Our experiments (summarized in Table 2) suggest that the syntactic information used for CCG supertagging affects agreement attraction patterns in word prediction only modestly, and, surprisingly, does not have a clear effect in the experiments most tied to syntactic effects on agreement attraction. In this section, we will discuss both why supertagging did not impact our models in the way we expected, as well as how we could build models that better capture the syntactic factors modulating agreement processing.

4.1.1 Why didn't supertagging lead to better simulations of syntactic experiments?

The error patterns corresponding to the contrasts that are most closely tied to syntactic structure—PP vs. RC (Bock & Cutting, 1992) and linear vs. syntactic distance (Franck et al., 2002)—were not more human-like in LM+CCG than LM-ONLY. This can be explained in two ways: either the supertagging objective did not give rise to sophisticated, human-like representations, or the representations supertagging gave rise to are not recruited for word prediction. We found support for the second explanation by analyzing how effectively the

Effect in Humans	LM-ONLY	LM+CCG	LM+CCG More Human-like?
Bock and Cutting (1992)			
PP > RC	x	No Difference	
Number Asymmetry	✓	Larger Effect	✓
Franck et al. (2002)			
Syntactic Distance > Linear Distance	x*	No Difference	
Haskell and Macdonald (2005)			
Linear Distance	✓	Larger Effect	✓
Humphreys and Bock (2005)			
Notional Number	x	No Difference	
Parker and An (2018)			
Core vs Oblique Argumentss.	x	No Difference	
Attraction in Grammatical Sentences	✓	Smaller Effect	✓
Wagers et al. (2009)			
Attraction from outside of RC	✓	No Difference	
Grammaticality Asymmetry	✓	No Difference	

Table 2: A summary of the experiments we simulated and the effects we found within LM-ONLY and LM+CCG models. The LM-ONLY column indicates whether we found a significant effect in the same direction as the original studies' authors found, and the LM+CCG column indicates whether we found a significant interaction between the relevant effect and the addition of CCG supertagging training as well as the direction of that interaction. *An effect is found in the LM-ONLY simulation of Franck et al. (2002), but in direction opposite of the effect found in humans.

efferent weights leading from a particular unit to each of the nodes in the output layers (words or supertags) distinguished between those nodes; this analysis showed that a given unit's efferent variance for supertag prediction—a proxy for the unit's sensitivity to syntax—was *inversely* correlated with the unit's efferent variance for next word prediction. Note that these two explanations are not mutually exclusive: It may be the case that the representations constructed by the model to perform supertagging are both unsophisticated and unused for word prediction; if those representations are not recruited for word prediction, we are simply unable to assess them using a paradigm based on a model's word prediction behavior.

While the auxiliary syntactic objective did not make performance more human-like across the board, it also did not make model performance *less* human-like. In each case, performance either did not change significantly or, in three cases, became more human-like. We take this as evidence that the more human-like behavior of LM+CCG models are not due just to random variation in the optimization process: if that was the case case we would expect changes in either direction with equal likelihood. Thus, despite a lack of significant changes in LM+CCG models' behavior on the specific, explicitly syntactic tasks we simulated, this pattern of results is consistent with the claim that additional pressure for models to learn syntactic properties of their input does lead to more human-like behavior broadly. Why supertagging did (or did not) help in a particular experiment, however, is unclear; we leave this question to future work.

4.1.2 How can we create models with more human-like syntactic processing?

Auxiliary training objectives are, at least in principle, an attractive tool, for a number of reasons: they can be implemented with minimal modification to model architecture; we can verify that the model has encoded the relevant information by monitoring its performance on the objective; and the idea that the representations used in language processing are shaped by the competing needs of various linguistic tasks is cognitively plausible (see, for example, the influence of orthographic pressures on the phonological representations used to detect rhymes, Seidenberg & Tanenhaus, 1979). Our negative results suggest, however, that auxiliary training objectives, at least as implemented in the present article, may not be a sufficiently effective tool for aligning the syntactic processing behavior of neural networks and humans.

How can we create models whose agreement error patterns show a human-like sensitivity to hierarchical structure? One potential path forward is to attempt to increase the pressure on LSTM-based models trained

on multiple objectives, such as the ones we used here, to transfer to one task the syntactic representation they acquire when learning to perform another. This pressure can be generated in a variety of ways. For example, reducing the size of the representations used by the models' LSTM encoders may make it less viable for the model to dedicate units to only one of the two objectives. Exposing models to more varied and complex training data, where local syntactic information is more critical for word prediction, may also incentivize the model to transfer supertagging-based syntactic representations to the word prediction task.

As an alternative approach, we could abandon auxiliary training objectives altogether and, instead, consider architectures that condition word prediction more directly on syntactic representations. The Recurrent Neural Network Grammar (Dyer, Kuncoro, Ballesteros, & Smith, 2016) architecture, for example, acts as a language model, but constructs explicit syntactic parses of its input during processing. This structure encourages the model to learn how best to use the hierarchical information contained in those parses to predict upcoming words. Prior work evaluating the syntactic abilities of these models have found them to be substantially better than LSTMs at predicting measures of processing difficulty in humans (Hale, Dyer, Kuncoro, & Brennan, 2018). Transformer architectures (Vaswani et al., 2017), like the GPT-2 model we evaluated, have also displayed significantly stronger syntactic abilities than LSTMs, particularly when trained on very large datasets (Hu et al., 2020). Transformers have been argued to implement processes akin to cue-based memory retrieval (Ryu & Lewis, 2021), a mechanism which is widely used to explain phenomena in agreement processing, as well as sentence processing more broadly (Lewis, Vasishth, & Dyke, 2006; Badecker & Kuminiak, 2007; Wagers et al., 2009; Parker & An, 2018). While our simulations using the transformer-based GPT-2 did not produce error patterns substantially closer to humans than LSTMs, we only explored a single transformer model, and thus a more thorough investigation of transformers — and the inductive biases inherent to that architecture — may show promise. As shown by our GPT-2 simulations, these architectures may also help address our models' high overall error rate. While this could simply be due to the additional power of larger models, it is likely that models with more explicitly structural representations will not only fail in ways that reveal their syntactic sophistication (as our agreement error analyses attempt to identify), but will use those representations to correctly identify the subject's number (in spite of attractors) more often.

4.2 Do the models learn consistent syntactic behavior from different types of training data?

Our comparison of LM-ONLY and LM+CCG models, discussed in the previous section, was based on models that were trained on a concatenation of two corpora: the Penn Treebank and a subset of Wikipedia. We also trained LM-ONLY models on each of these corpora separately, with the goal of evaluating the impact of dataset size and genre on syntactic behavior. We found that models trained solely on Wikipedia exhibited more human-like agreement error patterns when tested on PP and RC attractors than those trained on the Penn Treebank. We also found that models trained on the Penn Treebank agreed with the closer disjunct much more often than models trained on Wikipedia, making the Penn Treebank models closer to human behavior. This pair of findings indicates that models' syntactic processing behavior, as measured by their error patterns, is sensitive not only to the size of their training data, but also the genre.

For the purposes of using neural network language models as cognitive models, this sensitivity to small perturbations in training data is potentially worrying: If models are not sufficiently robust to variation in training data, the particular composition of the training dataset used becomes a critical part of our cognitive model's assumptions. The English Wikipedia corpus, though representative of a particular variant of English, is not representative of either the data observed by a child acquiring language, or of that observed by the average native speaker. This is also true of the Penn Treebank, which is composed primarily of financial news articles drawn from the Wall Street Journal. There are two major approaches we can take to address this problem: First, we could attempt to ensure that models trained for the purposes of cognitive modeling are trained on datasets that closely approximate the data a child would learn from, such as the CHILDES child-directed speech corpus (MacWhinney, 2000), though this particular corpus, which consists of conversations with very young children, is likely to be insufficient for a model intended to mimic adult processing of syntactically complex constructions (which tend to be rare in child-directed speech). The second approach would involve building models with stronger inductive biases that constrain the amount of variation that can be caused by the input data. While the supertagging objective may have weakly constrained the types of

solutions our models could find during training, stronger architectural inductive biases, like those imposed in models like Recurrent Neural Network Grammars (Dyer et al., 2016), may increase robustness to variation in training data.

4.3 Which linking function should we use to model agreement processing?

To turn language models into psycholinguistic models of agreement processing, we needed a method that converts the language model's output to a format that is comparable to the results of the human sentence completion or self-paced reading paradigm. Prior work has done so in two ways that are distinct from the ONE-SAMPLE linking function we described in Section 2.3. Here we contrast our method with these alternatives and provide a psycholinguistic interpretation of one class of potential linking hypotheses.

By contrast with the present study, Linzen and Leonard (2018) eschewed the use of a word prediction model entirely. Instead they trained their neural network as a verb number classifier: the decoder directly predicts the number feature of the verb from the preamble. This technique has two major limitations: First, it requires training data that is annotated with the number and position of the verb. From a cognitive perspective, such annotations are unlikely to be available to human learners; from a practical perspective, it is very costly to produce these annotations manually, and unreliable to do so automatically. Second, this training method prevents the model from learning syntactic constraints other than agreement and from using its knowledge of those other constraints to better predict agreement patterns. This contrasts with language models, which are incentivized to build representations for any property that might help it predict the following word from a sequence. Those representations are available to the model when it predicts the verb, and thus the verb's number. The only training signal available to a number classifier is whether or not it predicts the following verb's number correctly, and thus such a model is not incentivized to build representations for any other linguistic properties, including those that might interact with agreement in agreement attraction contexts.

Another linking function found in prior work is the one we will refer to as MAX-PROB, introduced by Linzen et al. (2016). Under this paradigm, a candidate pair of the singular and plural forms of a verb is selected, and the probabilities assigned by the language model to the two forms are compared. The model is evaluated as if it had produced the form whose probability is higher, regardless of the magnitude of the difference between the probabilities of the two forms.

The ONE-SAMPLE method we use preserves certain features of MAX-PROB. Like MAX-PROB, ONE-SAMPLE selects a candidate singular/plural pair of verbs prior to the selection of the verb's number feature. This design choice can be seen as a reflecting two sequential stages posited by some theories of language production (Bock & Levelt, 1994; Levelt, Roelofs, & Meyer, 1999): first, lemma selection—the selection of the word's canonical, morphologically unmarked form; and second, grammatical encoding, where grammatical features, like number, are marked. In our case, the word is first selected absent any number features, and then the number feature is determined and morphologically encoded onto the word. Under this interpretation, the model plus linking function combination presented here aims to capture only the second stage: grammatical encoding. The candidate pair we select in advance thus reflects the output of the lemma selection process.

The main difference between MAX-PROB and ONE-SAMPLE is that ONE-SAMPLE selects the output form probabilistically, with the probability of a singular form proportional to the probability assigned to the singular candidate by our language model. This gives ONE-SAMPLE one major advantage over MAX-PROB: It is sensitive to differences in probability between the verb forms in the language model, thereby capturing subtle effects that would be obscured if we used the MAX-PROB linking function.

An additional consequence of the choice to sample the model's output, rather than choose the more likely form, is that our models now exhibit non-deterministic behavior for a particular experimental item. Under MAX-PROB, a model that assigned a probability of 51% to the grammatical form would be taken to consistently produce the correct form of the verb. By contrast, under ONE-SAMPLE such a model would be only slightly above chance at producing the grammatical form of the verb. This is true even when the margin between the correct and incorrect forms' probabilities is large: a model that assigns 80% probability to the grammatical form would still produce errors in one out of five simulated trials when given the same preamble. This stochasticity better reflects the non-deterministic nature of human agreement errors—we would not expect a participant to always or never make errors on a particular item, but rather make an error on that item with some probability.

The difference between MAX-PROB and ONE-SAMPLE can be viewed as a reflection of the competence-performance distinction (Chomsky, 1965). The goal of MAX-PROB-based analyses is to determine whether a model *knows* that the form of the verb that agrees with the subject is better than the form that does not, and thus has acquired the linguistic *competence* associated with agreement. By contrast, our goal is to construct a model that makes the same errors in *performance* as humans. Thus we use our ONE-SAMPLE method, which provides an account of our cognitive model's production as a sample from the probability distribution provided by a language model (an account of the often erroneous *performance* of a speaker) rather than the MAX-PROB method. These two linking hypotheses lie at two ends of a spectrum of potential modeling assumptions: Under a paradigm where we take n samples from the distribution over the candidate pair provided by our language model and select the form sampled most often, ONE-SAMPLE is the case where we are limited to a single sample, while MAX-PROB matches the behavior in the limit as n approaches infinity. Future work might explore fitting n to human data, or comparing various choices of n to human behavior under various amounts of time pressure or memory load. For instance, one might expect that under high time pressure, human behavior might match an n closer to 1, while in an untimed proofreading task, behavior might match much higher values of n.

Modifications to ONE-SAMPLE may also help address some of the ways in which our models fail to match human behavior. For example, models based on ONE-SAMPLE will often assign significant probability mass to the form of the verb that the language model judges as less likely, which results in the high agreement error rates we observe in our simulations. This contrasts with MAX-PROB models, which assign no probability mass to the less likely form and thus, as discussed above, are insensitive to the underlying language model's level of certainty. Selecting a linking hypothesis that lies between these two extremes may lead to the best of both worlds, simultaneously preserving ONE-SAMPLE's sensitivity and reducing the overall rate of agreement errors. We leave an investigation of alternative linking functions for future work.

4.4 What can neural networks contribute to the the study of human syntactic processing?

Most psycholinguistic modeling, including in the area of agreement processing, adopts a cognitive process modeling approach—models are hand constructed, and consist of a number of interpretable, primitive cognitive operations organized sequentially (Gregg & Simon, 1967); each of these operations may have a small number of parameters that are fit to behavioral data. These models have, as their primary benefit, the ability to implement specific psycholinguistic hypotheses about the phenomena in question.

By contrast, neural networks are, on their face, black boxes (McCloskey, 1991). While we can attempt to modulate their behavior through the choice of their architecture and training task (or tasks), the mechanisms implemented by the model are learned from data during training. For a psycholinguist, this is a double-edged sword: It prevents us from testing a specific algorithmic theory like one could with a cognitive process model, but it also allows the model to develop solutions that one may not have otherwise considered. Because of this, neural network models can be used to evaluate claims in terms of relevant inductive biases; the processing mechanisms evolve over the course of learning through optimizing word prediction accuracy. In this work, we asked whether adding explicit bias toward more sophisticated syntactic representations would lead models to make more human-like agreement errors. By comparing models with and without that additional pressure, we could address that question, and determine whether strong syntactic representations were sufficient to explain the human patterns of agreement errors. Crucially, this was done without committing to a particular agreement mechanism, and without losing broad coverage: both types of models could be used to simulate agreement in any construction.

Another benefit of neural network modeling is that the mechanisms employed by neural networks are also necessarily *learnable* solutions; If our training task is ecologically valid, and our data is comparable to data a human might be exposed to, any solution developed by the model is, given the inductive biases assumed by our model choice, learnable from the input (Rumelhart & McClelland, 1987, among others). This is in contrast to traditional cognitive process models, in which it is often unclear how humans come to possess the hypothesized mechanism. Our current approach is based on training models to predict the next word over large natural corpora. Given the wealth of evidence that humans do something akin to word prediction during sentence processing (see Kutas, DeLong, & Smith, 2011 for a review), we take word prediction as a reasonable choice of training task (Elman, 1990). Our training data does, however, present two issues that

complicate the analogy to human learning: First, the corpora we use are not comparable to the input that a child would have access to when acquiring language. Future work attempting to strengthen the learning argument could consider using corpora of child-directed speech (i.e., CHILDES, MacWhinney, 2000) to evaluate whether less linguistically complex training data leads to similar behavior. Second, we must ensure that the amount of the data our models receive is comparable to that needed by humans to achieve a similar set of behaviors. In the long term, this perspective suggests considering all processing phenomena from the perspective of acquisition: can we construct a model that captures the relevant phenomena at the same stage of "acquisition" as human children?

Learnability considerations aside, a critic may still argue that the syntactic processing mechanisms models like ours learn are still insufficiently *explanatory*. The model's predictions are generated by a series of ostensibly uninterpretable matrix operations, a critic may say, so in calling a neural network model a model of language processing, we've replaced one black box, a human participant, with another. This issue is, however, not insurmountable. Unlike human participants, the inner workings of a neural network model can be recorded, probed, ablated, and inspected in a variety of other ways with little difficulty and with fewer ethical concerns, allowing researchers to approximate high-level, more easily interpretable operations that are implemented by a particular neural network (See, for example, Lakretz et al., 2019; Hupkes, Veldhoen, & Zuidema, 2018).

Thus, the limitations of neural networks do not indicate that they cannot contribute to psycholinguistics. Instead, they highlight the complementary nature of traditional psycholinguistic modeling and neural network methods. A neural network model that captures all of the phenomena we tested would not obviate the need for a traditional process model of agreement and agreement errors. Instead, such a model, and the set of inductive biases necessary and sufficient to construct it, would aid in the development of a process model. In particular, neural network models allow theories on the level of inductive biases to be instantiated as broad coverage, quantitative models of processing where underspecified aspects of models are learned from data. This creates a balance between two aspects of our model: Those that are specified and those that are learned from data. By inspecting exactly what we must specify in order to capture human patterns of behavior, we can better understand the core architectural features of our language processing faculties that lead to that behavior: Do we need biases toward certain kinds of syntactic representations to learn mechanisms that demonstrate human agreement behaviors?

We began by asking what agreement phenomena our LM-Only models can simulate, investigating what behavior a simple linear sequence learner with no biases toward hierarchical syntactic representations exhibits after being trained on word prediction. We then compare this model's agreement error patterns to a model with an explicit syntactic training objective, constraining the representations our model learns such that those representations must be sufficient to predict CCG supertags. Continuing to pursue approach by analyzing models with stronger and stronger syntactic inductive biases allows for a bottom-up approach to understanding phenomena like agreement attraction parallel to traditional hypothesis building. We first find the right inductive biases necessary for neural models to capture human performance through this exploration in the hypothesis space, and then analyze successful models using techniques from neural network to construct specific mechanistic hypotheses about the phenomena. These mechanistic hypotheses then serve to connect the particular innate or external biases and constraints that characterized our neural network model with traditional psycholinguistic models of the representations and processes that govern language processing. Though this work is just a preliminary step in this direction, and, as we discuss throughout this section, emblematic of many of the difficulties this direction poses, we see neural network models of language as a vital source of evidence into the nature of language processing mechanisms.

5 Conclusion

In this paper, we have proposed a framework for using neural language models to model human syntactic processing, and used that framework to evaluate the ability of a variety of neural language models with different training data and training objectives to simulate results from the agreement attraction literature. We aim to answer three questions: What behaviors do simple LM-Only models learn? Do LM+CCG models, with explicit syntactic supervision, perform in a more human-like way? Does the size and genre of the models' training corpus influence syntactic behavior?

Our simulations leave us with a few key findings: (1) neural network language models can capture a number of syntactic agreement effects, including linear distance effects, the grammaticality asymmetry and the number asymmetry; (2) an additional, explicitly syntactic training objective can lead models to encode additional syntactic information, but that information is not always used for word prediction; and (3) the ability of a language model to capture agreement phenomena is dependent not only on the inductive biases imbued by the models' architecture and training objectives, but also the size and composition of its training data.

More broadly, we see this work as the first step in constructing a neural network-based approach to modeling and understanding online agreement processing, and human syntactic processing more broadly. Under this approach, we first attempt to characterize the inductive biases necessary for matching human performance, then analyze the sufficiently biased, human-like models to generate detailed and testable hypotheses to be tested in humans. Crucially, this "bottom-up" approach is complementary to the cognitive process modeling approaches that are currently standard in psycholinguistics, as the issues inherent in cognitive process modeling (learnability and broad coverage analysis) can be addressed by using neural network approaches to generate and test statistically learned hypotheses. The work presented here works toward completing the first stage, helping characterize the inductive biases necessary to match human syntactic processing and evaluating a method for imbuing models with one such bias.

6 Acknowledgements

This work was supported by the United States–Israel Binational Science Foundation (award no. 2018284).

References

Arehalli, S., & Linzen, T. (2020). Neural language models capture some, but not all, agreement attraction effects. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society* (p. 370—376).

Badecker, W., & Kuminiak, F. (2006). Morphology, agreement and working memory retrieval in sentence production: Evidence from gender and case in Slovak. *Journal of Memory and Language*, *56*(1), 65-85. doi: 10.1016/j.jml.2006.08.004

Badecker, W., & Kuminiak, F. (2007). Morphology, agreement and working memory retrieval in sentence production: Evidence from gender and case in slovak. *Journal of memory and language*, *56*(1), 65–85.

Bangalore, S., & Joshi, A. (1999). Supertagging: An approach to almost parsing. *Computational Linguistics*, *25*(2), 237–265.

Bender, E. M., & Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of 58th Annual Meeting of the Association for Computational Linguistics.*

Bhatt, G., Bansal, H., Singh, R., & Agarwal, S. (2020). How much complexity does an rnn architecture need to learn syntax-sensitive dependencies? *arXiv preprint arXiv:2005.08199*.

Bock, K., & Cutting, J. C. (1992). Regulating mental energy: Performance units in language production. *Journal of Memory and Language*, *31*(1), 99–127. doi: 10.1016/0749-596X(92)90007-K

Bock, K., & Levelt, W. J. (1994). *Language production: Grammatical encoding.* Academic Press.

Bock, K., & Miller, C. A. (1991). Broken agreement. *Cognitive Psychology*, *23*(1), 45–93. doi: 10.1016/0010-0285(91)90003-7

Bock, K., Nicol, J., & Cutting, J. (1999). The Ties That Bind: Creating Number Agreement in Speech. *Journal of Memory and Language*, *40*(3), 330–346. doi: 10.1006/jmla.1998.2616

Bruni, E., Tran, N.-K., & Baroni, M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, *49*, 1–47.

Chen, D., & Manning, C. (2014, October). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 740–750). Doha, Qatar: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D14-1082` doi: 10.3115/v1/D14-1082

Chomsky, N. (1965). *Aspects of the theory of syntax.* MIT Press.

Clark, S. (2002). Supertagging for combinatory categorial grammar. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks (TAG+ 6)* (pp. 19–24).

Dyer, C., Kuncoro, A., Ballesteros, M., & Smith, N. A. (2016). Recurrent Neural Network Grammars. In *Proceedings of the 2016 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* San Diego, California: Association for Computational Linguistics.

Eberhard, K. M. (1999). The accessibility of conceptual number to the processes of subject–verb agreement in English. *Journal of Memory and Language*, *41*(4), 560-578. doi: https://doi.org/10.1006/jmla.1999.2662

Eberhard, K. M., Cutting, J. C., & Bock, K. (2005). Making Syntax of Sense: Number Agreement in Sentence Production. *Psychological Review*, *113*(3), 531–559. doi: 10.1037/0033-295X.112.3.531

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, *7*(2), 195–225.

Enguehard, É., Goldberg, Y., & Linzen, T. (2017). Exploring the syntactic abilities of rnns with multi-task learning. In *Proceedings of the 21st Conference on Computational Natural Language Learning* (pp. 3–14).

Ferrari, S., & Cribari-Neto, F. (2004). Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, *31*(7), 799–815.

Foppolo, F., & Staub, A. (2020). The puzzle of number agreement with disjunction. *Cognition*, *198*, 104161. doi: https://doi.org/10.1016/j.cognition.2019.104161

Franck, J., Lassi, G., Frauenfelder, U. H., & Rizzi, L. (2006). Agreement and movement: A syntactic analysis of attraction. *Cognition*. doi: 10.1016/j.cognition.2005.10.003

Franck, J., Vigliocco, G., & Nicol, J. (2002). Subject-verb agreement errors in French and English: The role of syntactic hierarchy. *Language and Cognitive Processes*, *17*(4), 371–404. doi: 10.1080/01690960

Futrell, R., Wilcox, E., Morita, T., Qian, P., Ballesteros, M., & Levy, R. (2019, June). Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 32–42). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/N19-1004` doi: 10.18653/v1/N19-1004

Gregg, L., & Simon, H. (1967). Process models and stochastic theories of simple concept formation. *Journal of Mathematical Psychology*, *4*(2), 246-276. doi: https://doi.org/10.1016/0022-2496(67)90052-1

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., & Baroni, M. (2018). Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1195–1205). New Orleans, Louisiana: Association for Computational Linguistics. doi: 10.18653/v1/N18-1108

Hale, J. (2001). A probabilistic earley parser as a psycholinguistic model. In *Second meeting of the north American chapter of the association for computational linguistics.*

Hale, J., Dyer, C., Kuncoro, A., & Brennan, J. (2018). Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (pp. 2727–2736). Melbourne, Australia: Association for Computational Linguistics. doi: 10.18653/v1/P18-1254

Hammerly, C., Staub, A., & Dillon, B. (2019). The grammaticality asymmetry in agreement attraction reflects response bias: Experimental and modeling evidence. *Cognitive Psychology*, *110*, 70–104. doi: 10.1016/j.cogpsych.2019.01.001

Haskell, T. R., & Macdonald, M. C. (2005). Constituent Structure and Linear Order in Language Production: Evidence From Subject-Verb Agreement. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *31*(5), 891-–904. doi: 10.1037/0278-7393.31.5.891

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hockenmaier, J., & Steedman, M. (2007). Ccgbank: a corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, *33*(3), 355–396.

Hu, J., Gauthier, J., Qian, P., Wilcox, E., & Levy, R. (2020). A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association*

for Computational Linguistics (pp. 1725–1744). Online: Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.158

Humphreys, K. R., & Bock, K. (2005). Notional Number Agreement in English. *Psychonomic Bulletin & Review*, *12*(4), 689–695.

Hupkes, D., Veldhoen, S., & Zuidema, W. (2018). Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artifical Intelligence Research*, *61*(1), 907–926.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Keung, L.-C., & Staub, A. (2018). Variable agreement with coordinate subjects is not a form of agreement attraction. *Journal of Memory and Language*, *103*, 1–18.

Kutas, M., DeLong, K. A., & Smith, N. J. (2011). A look around at what lies ahead: Prediction and predictability in language processing. In *Predictions in the brain: Using our past to generate a future* (p. 190-207). Oxford University Press.

Lakretz, Y., Desbordes, T., King, J., Crabbé, B., Oquab, M., & Dehaene, S. (2021). Can rnns learn recursive nested subject-verb agreements? *CoRR*, *abs/2101.02258*. Retrieved from `https://arxiv.org/abs/2101.02258`

Lakretz, Y., Kruszewski, G., Desbordes, T., Hupkes, D., Dehaene, S., & Baroni, M. (2019). The emergence of number and syntax units in lstm language models. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 11–20).

Levelt, W. J. M., Roelofs, A., & Meyer, A. S. (1999). A theory of lexical access in speech production. *Behavioral and Brain Sciences*, *22*(1), 1–38. doi: 10.1017/S0140525X99001776

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, *106*, 1126–1177. doi: 10.1016/j.cognition.2007.05.006

Lewis, R. L., Vasishth, S., & Dyke, J. A. V. (2006). Computational principles of working memory in sentence comprehension. *Trends in Cognitive Sciences*, *10*(10), 447–454. doi: 10.1016/j.tics.2006.08.007

Linzen, T., Dupoux, E., & Goldberg, Y. (2016). Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, *4*, 521–535.

Linzen, T., & Leonard, B. (2018). Distinct patterns of syntactic agreement errors in recurrent networks and humans. In *40th annual conference of the cognitive science society.*

Lorimor, H., Bock, K., Zalkind, E., Sheyman, A., & Beard, R. (2008). Agreement and attraction in russian. *Language and Cognitive Processes*, *23*(6), 769–799.

MacWhinney, B. (2000). *The CHILDES project: The database* (Vol. 2). Psychology Press.

Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330.

Marvin, R., & Linzen, T. (2018, October-November). Targeted syntactic evaluation of language models. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1192–1202). Brussels, Belgium: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D18-1151` doi: 10.18653/v1/D18-1151

McCloskey, M. (1991). Networks and theories: The place of connectionism in cognitive science. *Psychological Science*, *2*(6), 387-395. doi: 10.1111/j.1467-9280.1991.tb00173.x

McCoy, R. T., Min, J., & Linzen, T. (2020). BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP* (pp. 217–227). Online: Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.21

Merkx, D., & Frank, S. L. (2021). Human sentence processing: Recurrence or attention? In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics* (pp. 12–22). Online: Association for Computational Linguistics. doi: 10.18653/v1/2021.cmcl-1.2

Momma, S., & Ferreira, V. S. (2019). Beyond linear order: The role of argument structure in speaking. *Cognitive Psychology*, *114*, 101228. doi: https://doi.org/10.1016/j.cogpsych.2019.101228

Parker, D., & An, A. (2018). Not all phrases are equally attractive: Experimental evidence for selective agreement attraction effects. *Frontiers in Psychology*, *9*(aug), 1–16. doi: 10.3389/fpsyg.2018.01566

Pearlmutter, N. J., Garnsey, S. M., & Bock, K. (1999). Agreement Processes in Sentence Comprehension. *Journal of Memory and Language*, *41*(3), 427–456. doi: 10.1006/jmla.1999.2653

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners.

Rumelhart, D. E., & McClelland, J. L. (1987). Learning the past tenses of English verbs: Implicit rules or parallel distributed processing? In *Mechanisms of language aquisition.* (pp. 195–248). Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc.

Ryu, S. H., & Lewis, R. (2021, June). Accounting for agreement phenomena in sentence comprehension with transformer language models: Effects of similarity-based interference on surprisal and attention. In *Proceedings of the workshop on cognitive modeling and computational linguistics* (pp. 61–71). Online: Association for Computational Linguistics. doi: 10.18653/v1/2021.cmcl-1.6

Seidenberg, M. S., & Tanenhaus, M. K. (1979). Orthographic effects on rhyme monitoring. *Journal of Experimental Psychology: Human Learning and Memory*, *5*(6), 546–554. doi: 10.1037/0278-7393.5.6.546

Smith, N. J., & Levy, R. (2013). The effect of word predictability on reading time is logarithmic. *Cognition*, *128*(3), 302–19. doi: 10.1016/j.cognition.2013.02.013

Steedman, M. (1987). Combinatory grammars and parasitic gaps. *Natural Language & Linguistic Theory*, *5*(3), 403–439.

van Schijndel, M., & Linzen, T. (2018). Modeling garden path effects without explicit hierarchical syntax. In *40th annual conference of the cognitive science society.*

van Schijndel, M., & Linzen, T. (2021). Single-stage prediction models do not explain the magnitude of syntactic disambiguation difficulty. *Cognitive Science*, *45*(6), e12988. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.12988` doi: https://doi.org/10.1111/cogs.12988

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., & Gomez, A. (2017). & polosukhin, i.(2017). attention is all you need. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 5998–6008.

Vigliocco, G., & Nicol, J. (1998). Separating hierarchical relations and word order in language production: Is proximity concord syntactic or linear? *Cognition*, *68*(1). doi: 10.1016/S0010-0277(98)00041-9

Wagers, M. W., Lau, E. F., & Phillips, C. (2009). Agreement attraction in comprehension: Representations and processes. *Journal of Memory and Language*, *61*(2), 206–237. doi: 10.1016/j.jml.2009.04.002

Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., & Bowman, S. R. (2020). BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, *8*, 377–392.

Warstadt, A., Singh, A., & Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, *7*, 625–641.

Wilcox, E., Levy, R., Morita, T., & Futrell, R. (2018, November). What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP* (pp. 211–221). Brussels, Belgium: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/W18-5423` doi: 10.18653/v1/W18-5423

Wilcox, E. G., Gauthier, J., Hu, J., Qian, P., & Levy, R. P. (2020). On the Predictive Power of Neural Language Models for Human Real-Time Comprehension Behavior. In *42nd annual conference of the cognitive science society.*

Williams, E. (1978). Across-the-board rule application. *Linguistic Inquiry*, *9*(1), 31-43.

A Edits to Experimental Items

The neural network models we train operate on the word level, and depend on the set of words contained in the models' training sets in order to learn word-level representations. When a model encounters a word it has not seen in training, it uses the representation of a special <UNK> token that replaces words that appear fewer than five times in the input.

Because most experimental manipulations depend on the features of a particular word, the experimental materials we use in our simulations must be edited so as to avoid <UNK> tokens preventing the models' from being able to interpret those features. Below, we will list, for each set of experimental materials, the

changes made to those materials to match the vocabulary of the Wikipedia dataset. Due the the significant vocabulary limitations of the Penn Treebank dataset, we provide a full list of the modified items. Since our goal is to replace rare words outside of the models' vocabularies with words that the models have observed, we note that the replacement words are necessarily higher frequency. We also do not control for subword properties (i.e., number of characters, etc.), since our LSTM models treat words as atomic units and thus have no access to those properties.

A.1 Modifications to match the Wikipedia Vocabulary

A.1.1 Bock and Cutting (1992)

We identified four subjects or attractors which did not have both their singular and plural form in our vocabulary. Below, we provide one condition (singular subject, singular attractor, PP modifier) of the edited items containing each of those noun phrases, with the noun appearing in the original items shown in parentheses.

(19) The performer (fire-eater) in the carnival show

(20) The inspector (superintendent) of the technical school

(21) The letter (memo) from the junior executive

(22) The lab (laboratory) with the analog computer

In addition, there were 3 words that were not in the Wikipedia training set that were not a part of the critical manipulation, and thus remained as <UNK> tokens during simulations. We provide example sentences containing those words below:

(23) The performer who <UNK> (enlivened) the show

(24) The neural zone around the <UNK> (arcturian) solar system

(25) The traffic jam on the <UNK> (Okemos) street

A.1.2 Franck et al. (2002)

All of the words used in the experimental materials were within the Wikipedia vocabulary with one exception, *innkeeper*. We provide a sample sentence of the item with *innkeeper*, and its replacement, *inn*:

(26) The meal for the guest of the inn (inn-keeper)

A.1.3 Haskell and Macdonald (2005)

A sample sentence for each item with changes is listed below:

(27) Ask Ronnie if the pearl (ruby) or the diamonds

(28) Do you remember if the table (dresser) or the beds

(29) Did Naomi say whether the shelf (bookshelf) or the beds

(30) Marcus will tell you whether the pitcher or the pots (teapots)

(31) Do you remember if the cocktail (martini) of the beers

(32) Find out whether the shovel or the buckets (rakes)

No <UNK> tokens in remained after these changes.

A.1.4 Humphreys and Bock (2005)

No words in the Humphreys and Bock (2005) experimental materials were not in the Wikipedia vocabulary, and thus no modifications were made to the items.

A.1.5 Parker and An (2018)

One word critical to the manipulation, *stewardess*, was replaced as so:

(33) The woman (stewardess) who sat the passengers certainly was very pleased with the long flight.

The adverb *unsurprisingly*, though not critical to the manipulation, was also not in the vocabulary. An example sentence with it replaced with an <UNK> token is provided below:

(34) The waitress who sat near the girl <UNK> (unsurprisingly) was unhappy about all the noise.

A.1.6 Wagers et al. (2009)

Two words, one critical to the manipulation and one not, were not in the Wikipedia vocabulary. An example item with both words is shown below:

(35) The vendor who the host (hostess) suggests to their friends are excellent but <UNK> (outrageously) expensive.

A.2 Penn Treebank Items

A.2.1 Bock and Cutting (1992)

1. The new tape from the popular rock artist
2. The newspaper from the British government agency
3. The performer in the carnival show
4. The bright light in Doctor Smith 's examination room
5. The security force at the giant manufacturing plant
6. The interview of the famous television host
7. The popular leader of the left dissident group
8. The teacher for the chemistry student
9. The inspector of the technical school
10. The letter from the junior executive
11. The neutral area around the <UNK> solar system
12. The traffic block on the <UNK> street
13. The office of the certified employee
14. The rebel in the dangerous conflict
15. The actor in the blockbuster film
16. The consultant for the growing firm
17. The teaching aide for the science lab
18. The employee with the diplomat 's message
19. The star of the <UNK> production
20. The corporation with the banking monopoly
21. The picture of the prominent politician

22. The writer of the modern book
23. The teacher with the special education certificate
24. The member at the union meeting
25. The director of the new motion picture
26. The candidate for the corporate promotion
27. The editor of the history book
28. The lab with the old computer
29. The activist at the political rally
30. The student in the Spanish class
31. The Peace Corps member in the African town
32. The leader of the Roman city state

A.2.2 Franck et al. (2002)

1. The ad from the office of the real estate agent
2. The announcement by the director of the foundation
3. The article by the writer for the magazine
4. The author of the speech about the city
5. The computer with the program for the experiment
6. The contract for the actor in the film
7. The dog on the path around the lake
8. The friend of the editor of the magazine
9. The gift for the daughter of the tourist
10. The helicopter for the flight over the hill
11. The lesson about the government of the country
12. The letter from the friend of my brother
13. The book by the developer of the machine
14. The chair on the deck of the ship
15. The gift for the guest of the hotel
16. The museum with the picture of the artist
17. The design for the engine of the plane
18. The payment for the service to the school
19. The photo of the girl with the baby
20. The post in the support for the platform
21. The prescription by the doctor from the clinic

22. The producer of the movie about the artist
23. The publisher of the book about the king
24. The setting for the movie about the scientist
25. The sign in the garden near the mansion
26. The switch for the light in the room
27. The message to the friend of the politician
28. The threat to the president of the company
29. The tour of the garden near the park
30. The train to the city on the lake
31. The truck on the bridge over the stream
32. The discussion about the topic of the paper

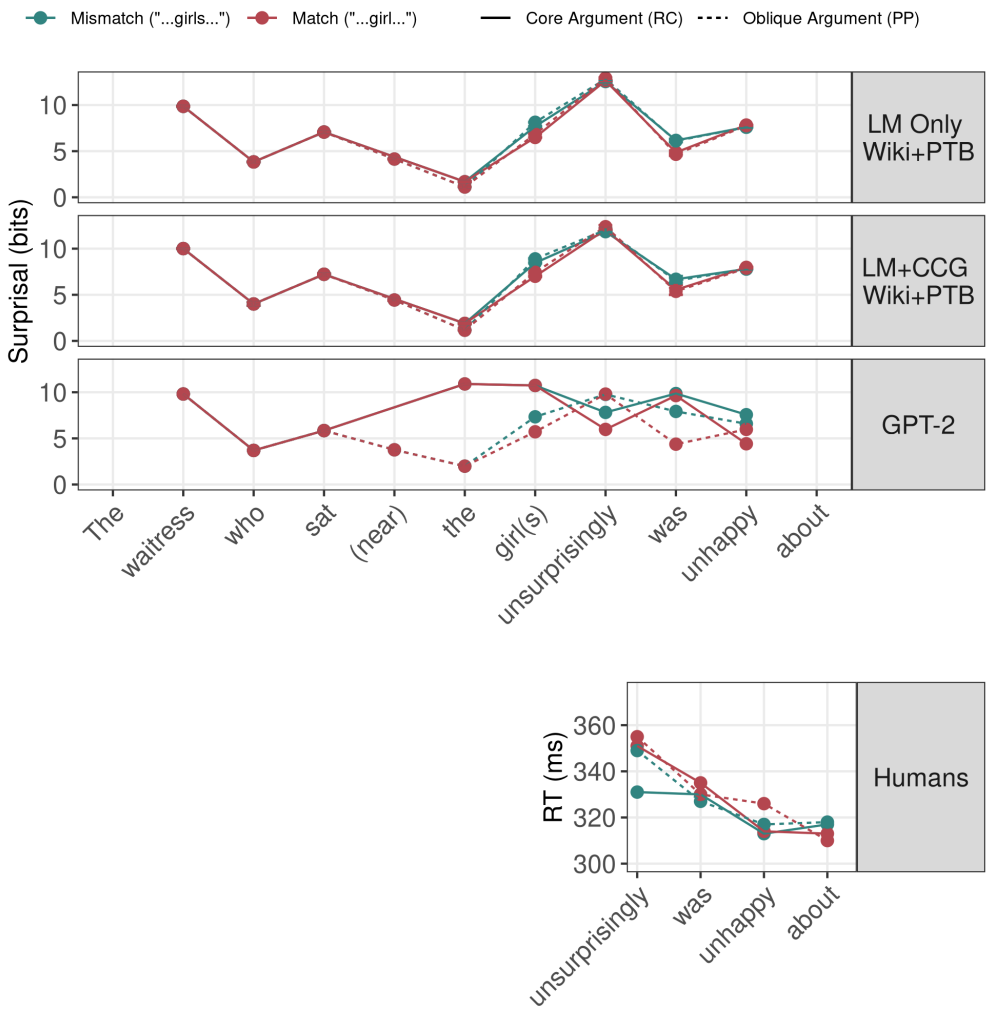
A.2.3 Haskell and Macdonald (2005)

1. Can you ask <UNK> if the kids or the adult
2. Do you know if the mice or the monitor
3. Do you think the soybeans or the apple
4. Have you heard whether the teachers or the principal
5. How do I know if the shelves or the floor
6. I <UNK> tell whether the doctors or the professional
7. Do the <UNK> say if the stores or the restaurant
8. We need to know if the potatoes or the grain
9. I want to know if the sheets or the color
10. I need to know if the tables or the chair
11. Maria probably knows if the photos or the painting
12. It didn't matter to me if the magazines or the book
13. It is hard to tell whether the steelmakers or the engineer
14. Ask <UNK> if the metals or the diamond
15. I wonder if the plants or the fly
16. It doesn't really matter whether the contractors or the bank
17. Can you tell me whether the swings or the court
18. Do you think the windows or the wall
19. Do you remember if the doors or the carpet
20. Did <UNK> say whether the book shelves or the desk
21. Can you ask the guide if the pencils or the gun

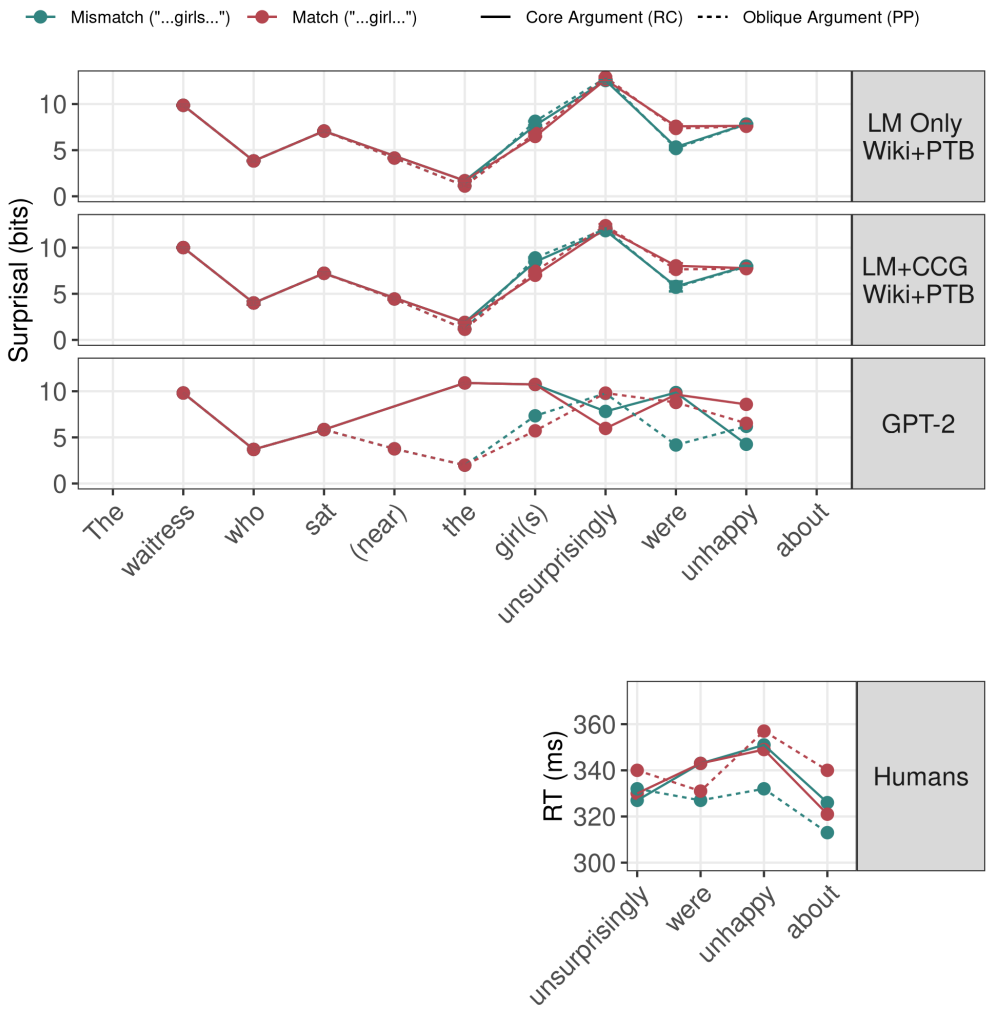
22. Did <UNK> say whether the lights or the plant
23. Can you tell me if the TVs or the phone call
24. Can you tell me whether the boxes or the can
25. The book must say whether the trails or the river bank
26. Would you say the fax machines or the printer
27. Ask the doctor whether the passengers or the driver
28. Marcus will tell you whether the pipelines or the road
29. Do you remember if the waters or the beer
30. Ask the boss if the cases or the box
31. <UNK> confused about whether the pictures or the prize
32. Do you think the lights or the sign
33. Find out whether the prices or the tax
34. Did you think the teams or the expert
35. Can you find out if the barrels or the package
36. Do you know whether the phones or the camera
37. The board wants to know if the theaters or the coffee shop
38. <UNK> must know whether the book stores or the restaurant
39. Can you tell me whether the brokers or the salesman
40. Tell me whether the boards or the president

B Full Sentence Surprisals for Comprehension Simulations

B.1 Parker and An (2018)



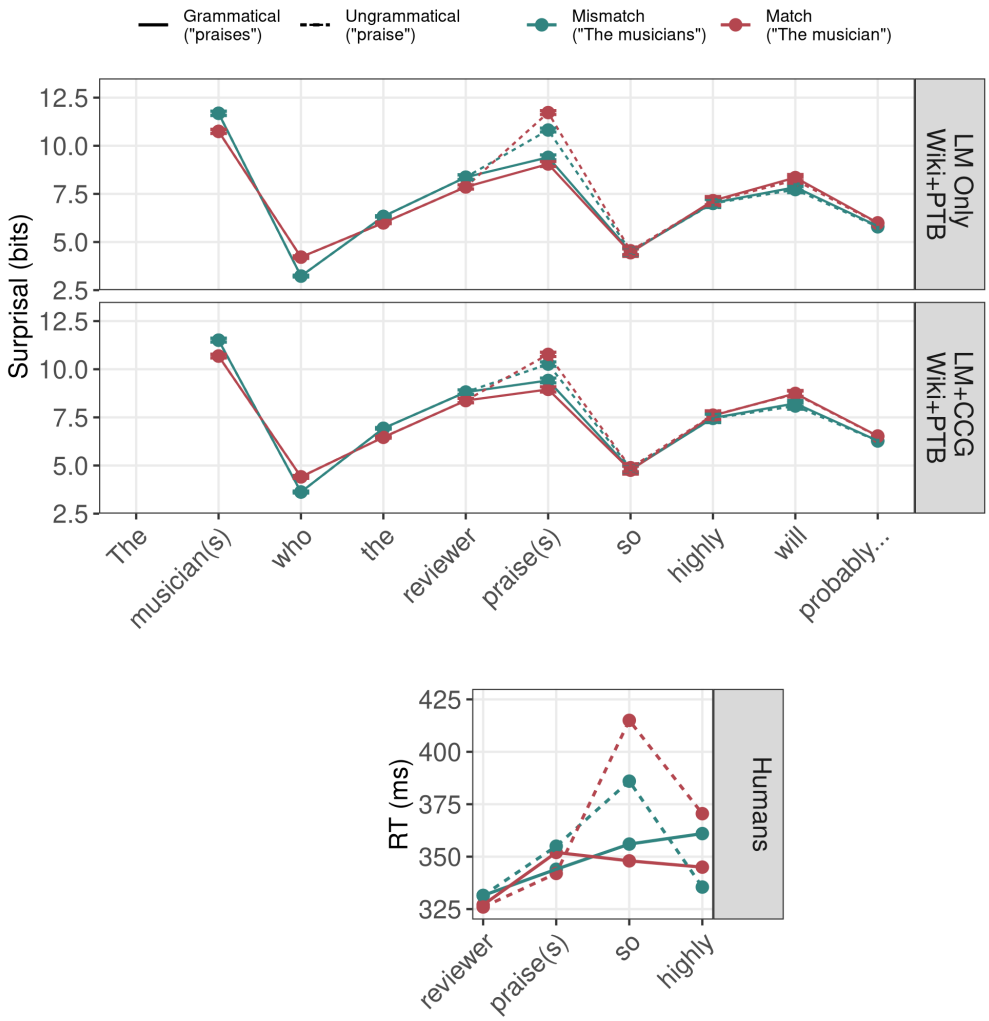
(a) Simulation Results, Grammatical Sentences



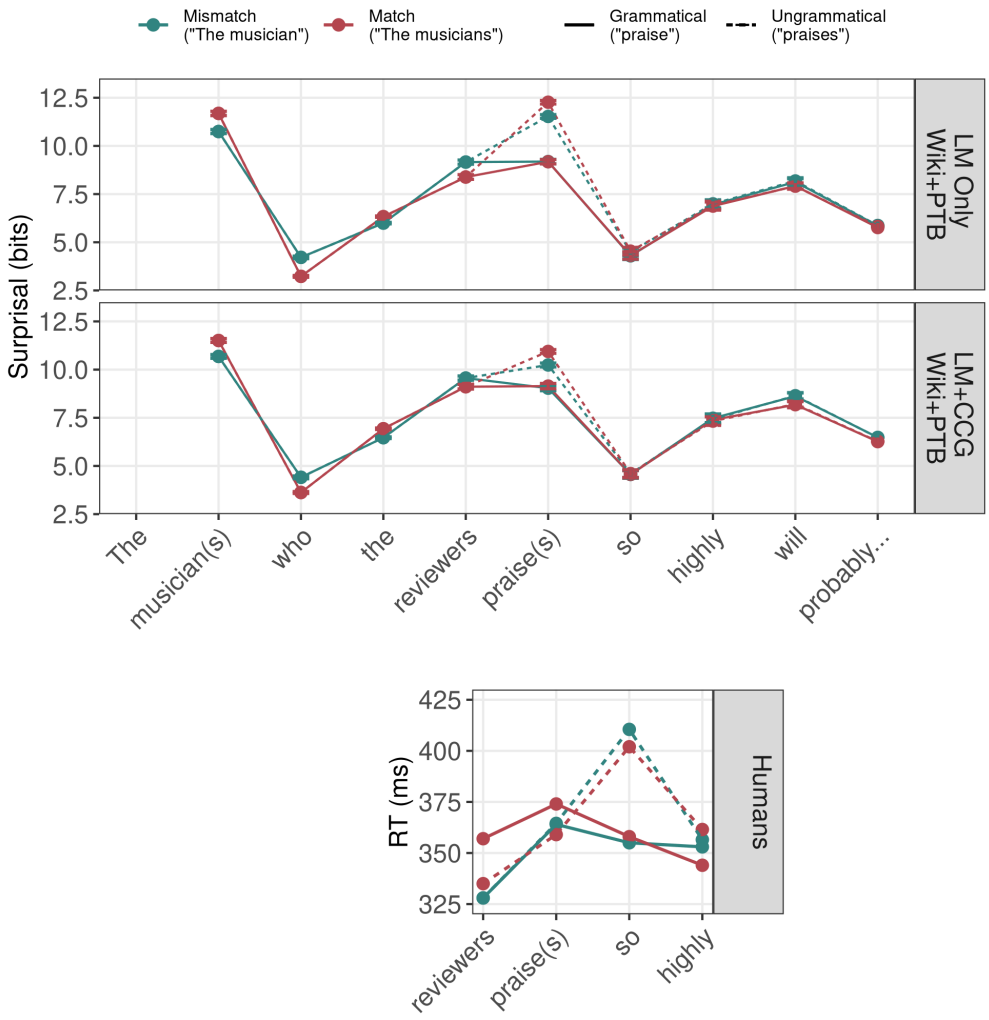
(b) Simulation Results, Ungrammatical Sentences

Figure 14: Word-by-word surprisals for models in our simulation of Parker and An (2018). Error bars are standard errors. Since models were given no context prior to the first word, no surprisal is given for the first word of the sentence (*The*). Since *near* only appears in the oblique argument condition, no surprisal is provided for the token in the core argument condition. The critical region here is at the verb *was/were*, where the grammaticality of the agreement relation becomes clear. If an attraction effect manifests in grammatical sentences, surprisal will be higher in the mismatch condition than for those in the mismatch condition. If such an effect manifests in ungrammatical sentences, surprisal will be lower in the mismatch condition than in the match condition.

B.2 Wagers et al. (2009)



(a) Simulation Results, Singular Subject



(b) Simulation Results, Plural Subject

Figure 15: Word-by-word surprisals for models in our simulation of Wagers et al. (2009). Error bars are standard errors. Since models were given no context prior to the first word, no surprisal is given for the first word of the sentence (*The*). The critical region here is at the verb *praise(s)*, where the grammaticality of the agreement relation becomes clear. If an attraction effect manifests in grammatical sentences, surprisal will be higher in the mismatch condition than for those in the mismatch condition. If such an effect manifests in ungrammatical sentences, surprisal will be lower in the mismatch condition than in the match condition.