Analysing Neural Language Models: Contextual Decomposition Reveals Default Reasoning in Number and Gender Assignment

Jaap Jumelet
jumeletjaap@gmail.com
University of Amsterdam

Willem Zuidema
w.h.zuidema@uva.nl
ILLC, University of Amsterdam

Dieuwke Hupkes
d.hupkes@uva.nl
ILLC, University of Amsterdam

Abstract

Extensive research has recently shown that recurrent neural language models are able to process a wide range of grammatical phenomena. *How* these models are able to perform these remarkable feats so well, however, is still an open question. To gain more insight into what information LSTMs base their decisions on, we propose a *generalisation* of *Contextual Decomposition* (GCD). In particular, this setup enables us to accurately distil which part of a prediction stems from semantic heuristics, which part truly emanates from syntactic cues and which part arise from the model biases themselves instead. We investigate this technique on tasks pertaining to syntactic agreement and co-reference resolution and discover that the model strongly relies on a *default reasoning* effect to perform these tasks.

1 Introduction

Modern language models that use deep learning architectures such as LSTMs, bi-LSTMs and Transformers, have shown enormous gains in performance in the last few years and are finding applications in novel domains, ranging from speech recognition and writing assistance to autonomous generation of fake news. Understanding how they reach their predictions has become a key question for NLP, not only for purely scientific, but also for practical and ethical reasons.

From a linguistic perspective, a natural approach is to test the extent to which these models have learned classical linguistic constructs, such as inflectional morphology, constituency structure, agreement between verb and subject, filler-gap dependencies, negative polarity or reflexive anaphora. An influential paper using this approach was presented by Linzen et al. (2016), who investigated the performance of an LSTM-based language model on number agreement. In many later papers (e.g. Gulordava et al., 2018; Wilcox et al., 2018; Jumelet and Hupkes, 2018; Marvin and Linzen, 2018; Giulianelli et al., 2018) a wide spectrum of grammatical phenomena has been investigated, assessing these grammatical abilities in a mainly "behavioural" fashion, by considering the model's output.

In this paper, we take it as established that neural language models have indeed learned a great number of non-trivial linguistic patterns and ask instead *how* language models come to show this behaviour, and, more specifically, what kind of information they use to come to their decisions. There exist already a number of approaches that look inside the high-dimensional vector representations and non-linear functions of these models, trying to track the flow of information. In the next section, we will review some of that work, distinguishing between hypothesis-driven and data-driven methods. We highlight in particular one method called Contextual Decomposition (CD, Murdoch et al., 2018), that combines the strengths of hypothesis- and data-driven analysis methods.

In the remainder of this paper, we then propose a generalisation of this method, which we call Generalised Contextual Decomposition ("GCD"). We derive equations for GCD for the case of a unidirectional (one or multi-layer) LSTM (Hochreiter and Schmidhuber, 1997), and use the method to analyse how a language model processes two different phenomena: number agreement and gendered pronoun resolution.

We demonstrate the power of GCD through the revelation of some important asymmetries in the way that both the singular-plural and the male-female distinction are handled. In particular, we find evidence for a *default reasoning* effect, which we believe could also be important for future work on detecting and removing bias: a default category (singular, masculine) appears to be hard-coded in

Proceedings of the 23rd Conference on Computational Natural Language Learning, pages 1–11
Hong Kong, China, November 3-4, 2019.

the weights of the language model, number and gender information in the word embeddings themselves mainly plays a role for phrases of the opposite category (plural, feminine). Furthermore, GCD enables us to investigate pronoun resolution in a way that has not been done before: by delving into the model reasoning we are able to accurately pinpoint where and how this resolution takes place.[1]

2 Network analysis methods

Recently, methods to open the blackbox of deep neural networks have become an important research area (see Poerner et al., 2018; Belinkov and Glass, 2019, for recent reviews of proposed methods in NLP). We distinguish between hypothesis-driven methods, and data-driven methods. Hypothesis-driven methods include probes or *diagnostic classifiers*, that test whether specific, a priori defined information can be decoded from the internal states of a neural model, many ablation studies, and types of correlation analysis, where correlations between the structure of internal representations of better and lesser understood models are studied). An example of this approach is Giulianelli et al. (2018), who trained linear diagnostic classifiers on all layers and gate activations of an LSTM to predict the number of the subject that the verb, occurring later in the sentence, needs to agree with (i.e. the number-agreement task). Their results show that the relevant information is encoded in a different way in different components of the model, and at different times while processing a sentence. This result is interesting, because it starts from a clearly interpretable hypothesis (number information must be maintained somewhere while the network traverses the sentence), but the work also demonstrates the limitations of the approach: It progresses one hypothesis about one linguistic pattern at a time and involves much training, work, and computation at each step.

Data-driven methods include gradient-based methods and contextual decomposition. An example of a gradient-based method is Arras et al. (2017), who adapt Layer-wise Relevance Propagation (LRP, Bach et al., 2015) to the case of LSTMs. The key idea is to run the LSTM on each input of interest (the forward pass), then define a relevance vector at the output layer and propagate that relevance backwards through the network. The relevance vector simply singles out the dimensions of the output of interest, and sets all other dimension to zero. The backward pass is almost standard backpropagation, except that relevance does not backpropagate into the gates.While Arras et al.'s results reveal interesting patterns in sentences used in a sentiment classification task, their work illustrates some limitations as well. In particular, the work deals with a classification task with few classes, aggregates relevance per word for each predicted class, but offers little insight in how word meanings interact to build up sentence meaning beyond 'pushing in the right direction' vs. 'pushing in the wrong direction'.

An alternative data driven method, and the one that we will expand on in this paper, is Contextual Decomposition for LSTMs (CD, Murdoch et al., 2018). The key idea behind this technique is to partition the hidden states into two components, that Murdoch et al. label 'relevant' and 'irrelevant'. For each word in a sentence, they do a forward pass that computes all cell and gate activations as in normal operation of the neural network, but also partition each activation value of each neuron in h or c in a part that is *caused* by some selected token or phrase in focus, and a part that is not. They achieve this by deriving a factorisation of the update formulas for h and c, that expresses them as a long sum of components and then selecting some of these components as being relevant, and others as irrelevant. Qualitative results on sentiment analysis suggest that CD can attribute roles to words in a sentence very well, better than alternatives the authors considered (which, unfortunately, did not include LRP).

CD thus requires no extra training and requires only the forward pass of the network. It can easily be extended to work efficiently with many classes, such as the language modelling task that we are interested in. In the next section, we will define CD more precisely, where we will use the terms *inside* and *outside* rather than relevant and irrelevant. We then propose a generalisation that allows us to experiment with different *hypotheses* on what goes into the *inside* and *outside* bins, enabling some of the advantages of hypothesis-driven analysis methods to be brought into this data-driven method.

[1] We have integrated all our code in `diagnnose` (Jumelet and Hupkes, 2019), a well-documented analysis library which facilitate the diagnosis of neural network activations: `github.com/i-machine-think/diagnnose`.

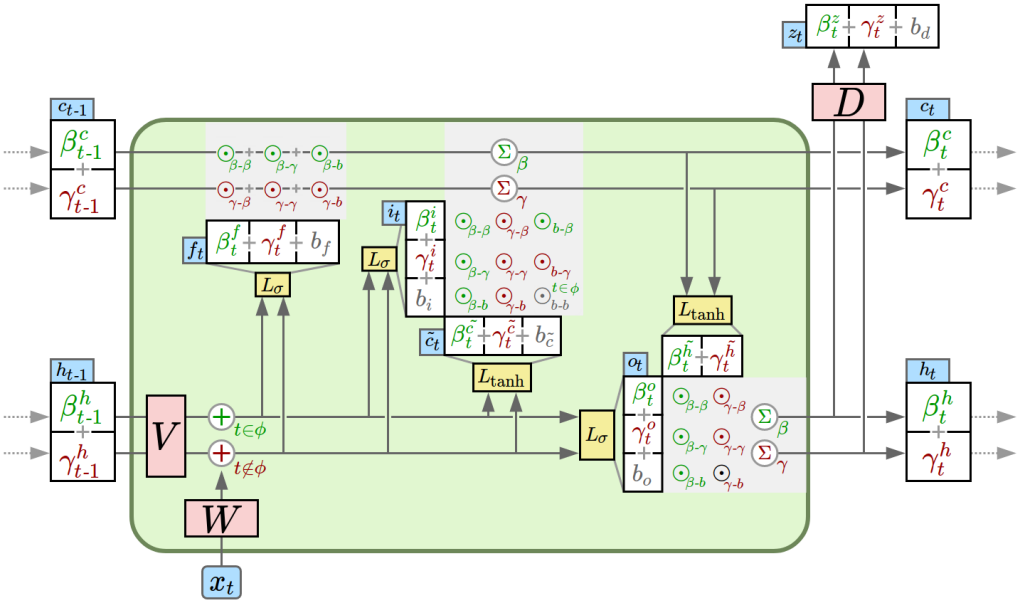


Figure 1: A graphical overview of GCD, based on the LSTM design of Olah (2015). ϕ denotes the phrase in focus, and $t \in \phi$ implies the action is only performed when step t is part of ϕ. $\odot$ denotes an individual interaction; green interactions are added the β part and red interactions to γ. V, W, and D represent the linear projections of the LSTM itself. The interaction set denoted here corresponds to the IN set of Equation 12.

3 Generalised Contextual Decomposition

In this particular study, we consider the LSTM language model that was made available by Gulordava et al. (2018). This language model (LM) is a 2-layer LSTM with 650 hidden units in both layers, trained on a corpus with Wikipedia data. Given the relevance of the specific LSTM-dynamics for the understanding of the main method of our paper, we repeat the equations that describe it below.

$$f_t = \sigma(W_f x_t + V_f h_{t-1} + b_f) \quad (1)$$
$$i_t = \sigma(W_i x_t + V_i h_{t-1} + b_i) \quad (2)$$
$$\tilde{c}_t = \tanh(W_{\tilde{c}} x_t + V_{\tilde{c}} h_{t-1} + b_{\tilde{c}}) \quad (3)$$
$$o_t = \sigma(W_o x_t + V_o h_{t-1} + b_o) \quad (4)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (5)$$
$$h_t = o_t \odot \tanh(c_t) \quad (6)$$
$$z_t = D h_t + b_d \quad (7)$$
$$p_t = \text{SoftMax}\,(z_t) \quad (8)$$

The final model output p_t represents a multinomial distribution over the model's vocabulary. Throughout the paper we refer to the bias terms b as the model *intercepts*, to avoid confusion with general biases that the model may have.

CD To compute the contributions of one or multiple input tokens (said to be *in focus*) to the output of an LSTM cell, Murdoch et al. (2018) divide each cell and hidden state into a sum of two parts: a β part, which contains the part of this particular state that stems from **inside** this phrase, and a γ part, which contains information coming from words **outside** this phrase. The output logit z_t can then be redefined as

$$\begin{aligned} z_t = D h_t + b_d &= D\beta_t^h + D\gamma_t^h + b_d \\ &= \beta_t^z + \gamma_t^z + b_d \end{aligned}$$

with β_t^z providing a quantitative score of the phrase's contribution to the logit. How a particular hidden state h_t is partitioned into β_t^h and γ_t^h is determined by two things: **i)** The decomposition of the *previous* states c_{t-1} (β_{t-1}^c and γ_{t-1}^c) and h_{t-1} (β_{t-1}^h and γ_{t-1}^h), and **ii)** Which *interactions* between the different β and γ terms, the intercepts b, and the input x_t are considered to be part of the inside contribution of the phrase. We provide a graphical overview of our setup in Figure 1.

Factorised activation functions The gate interactions cannot yet be expanded into a cross-term of their input parts, due to the non-linear activation that wraps them. Murdoch et al. define a method to factorise the sigmoid and tanh functions for each specific gate into a sum of contributions of the input terms, such that

$$\tanh(\sum_{i=1}^{N} y_i) = \sum_{i=1}^{N} L_{\tanh}(y_i)$$

$L_{\tanh}$ expresses the *contribution* of each input, which is computed by averaging over the differences of all possible permutations of the input

terms; a procedure that corresponds to the calculation of the Shapley values (Shapley, 1953).[2]

Before this factorisation is performed, an input token x_t is added to the inside part β if it is part of the phrase for which we decompose (i.e. the phrase *in focus*), otherwise it is added to γ. Equation 1, for example, can then be rewritten as:

$$\begin{aligned} f_t &= \sigma\left(V_f\beta^h_{t-1} + V_f\gamma^h_{t-1} + W_f x_t + b_f\right) \\ &= L_\sigma\left(V_f\beta^h_{t-1} + W_f x_t\right) + L_\sigma(\gamma^h_{t-1}) + L_\sigma(b_f) \end{aligned} \tag{9}$$

where x_t is considered to be inside the phrase in focus and therefore added to the β part (denoted in green for extra emphasis). A similar sum can be written down for the input gate i_t and the candidate cell state $\tilde{c}_t$. This allows the two products $f_t \odot c_t$ and $i_t \odot \tilde{c}_t$ of Equation 5 to be expanded into a sum of cross-terms between the decomposed gate and (candidate) cell values. Expanding the forget and input gate results in 15 cross-terms, that each express different interactions between the current input, previous β and γ terms, and the model intercepts.

Murdoch et al. state they observed improvements when the intercept term is fixed to the first position in each permutation. Consequently, however, these intercepts are assigned a relatively larger contribution, as their fixed position makes their contribution independent of the magnitudes of the other terms. We therefore pose that the full set of permutations should be considered, to assign unbiased contributions to each input term.[3]

Decomposing interactions Based on all the different interaction terms, the decomposition is determined by *which* of these interactions should be considered to belong to the inside part β of the next cell state and which to the outside part γ.

In the formulation of Murdoch et al., all interactions with outside parts γ_t are disregarded for the computation of β_{t+1}, and therefore only information directly stemming from the β_t terms with no interference from γ_t is taken into account. Of the 15 cross-product terms described above, this leaves 5 terms to be part of β^c_{t+1}:

$$\begin{aligned} \beta^c_{t+1} = {} & L_\sigma(V_f\beta^h_t + W_f x_t) \odot \beta^c_t && \beta\text{-}\beta \\ & + L_\sigma(b_f) \odot \beta^c_t && \beta\text{-}b \\ & + L_\sigma(V_i\beta^h_t + W_i x_t) \odot L_{\tanh}(V_{\tilde{c}}\beta^h_t + W_{\tilde{c}} x_t) && \beta\text{-}\beta \\ & + L_\sigma(V_i\beta^h_t + W_i x_t) \odot L_{\tanh}(b_{\tilde{c}}) && \beta\text{-}b \\ & + L_{\tanh}(V_{\tilde{c}}\beta^h_t + W_{\tilde{c}} x_t) \odot L_\sigma(b_i) && \beta\text{-}b \end{aligned} \tag{10}$$

The remaining 10 terms from the cross-product are put in γ^c_{t+1}. We use the notation $\{\beta\text{-}\beta,\ \beta\text{-}b\}$ to concisely describe this set of interactions. The decomposition of the hidden state is created by decomposing the output gate:

$$\begin{aligned} \beta^h_{t+1} = {} & L_\sigma(V_o\beta^h_t + W_o x_t) \odot \beta^c_{t+1} \\ & + L_\sigma(b_o) \odot \beta^c_{t+1} \end{aligned} \tag{11}$$

The decomposed contribution score β^z_T over the model vocabulary at step T of some phrase in focus is then calculated by passing the decomposed hidden state to the decoder, i.e. $D\beta^h_T$. This score can be expressed as a relative contribution by normalising it by the full model logit z (including b_d). In a multi-layer LSTM, β and γ parts are not only propagated *forward*, but also *upward*, where they are added to their respective parts in the layer above them. For initialisation β is set to a zero vector, and γ is set to the initial LSTM states.[4]

Generalising CD While Murdoch et al. (2018) consider only one way of partitioning interactions between inside and outside components, their setup can be quite easily generalised to also allow other interactions to be included in the inside terms β. To obtain a better insight into how different interactions contribute to the final prediction, we experiment with various ways of defining the set of relevant interactions.

A particular case concerns the interactions between β and γ. It wouldn't be correct to completely attribute the information flowing from these interactions to the phrase in focus, but disallowing any information stemming from interactions of a phrase with a subsequent token results in loss of relevant information. Consider, for instance, the verb prediction in a number agreement task. While the correct verb *form* depends only on the subject, the right *time* for this information to surface depends on the material in between, which in the setup described in Equation 10 would be discarded by assigning the β-γ interactions to γ.

[2] In the original formulation this procedure is called *linearizing*. We deemed this term to be slightly confusing, as the resulting functions L are still non-linear.

[3] We only discovered the impact of this decision after the paper had already been reviewed. While using the full set of permutations did, fortunately, not qualitatively change our conclusions, the exact numbers presented in this work thus differ from the earlier version of this paper. For completeness, we report the original results with the fixed intercept positions in the supplementary materials of this article.

[4] For the initial states we use the activations that follow from the short phrase ". <eos>". This phrase resets the model state to a clean slate, and leads to better results than using 0-valued activations.

Taking inspiration from Arras et al. (2017), and based on their motivation, we add the β_s-γ_g interaction to the relevant interaction set, while still disregarding a γ_s-β_g interaction. The g subscript denotes the part of the interaction that is coming from the gate, and s the source part. We denote this amended interaction as β-γ^*.

Furthermore, we follow the addition of Singh et al. (2019) of only adding the intercept interactions b-b to the inside part if the current time step is part of the phrase in focus, which we denote as $b\text{-}b \in x$. We add these β-γ^* and $b\text{-}b \in x$ interactions to Equation 10, resulting in the following decomposition that is presumed to come from **inside** the phrase in focus (denoted as IN):

$$\begin{aligned}\beta^c_{t+1} = \{&\beta\text{-}\beta, \beta\text{-}b\} \\ &+ L_\sigma(V_f\gamma^h_t) \odot \beta^c_t && \beta\text{-}\gamma^* \\ &+ L_\sigma(V_i\gamma^h_t) \odot L_{\tanh}(V_{\tilde{c}}\beta^h_t + W_{\tilde{c}}x_t) && \beta\text{-}\gamma^* \\ &+ L_\sigma(b_i) \odot L_{\tanh}(b_{\tilde{c}}) && b\text{-}b \in x \quad (12)\end{aligned}$$

We also experimented with various other interaction sets. To determine the influence of the gate intercepts, we create an interaction set that does not take the input embeddings into account at all: $\{\beta\text{-}\beta, \beta\text{-}\gamma^*, \beta\text{-}b, b\text{-}b\}$, with x always added to γ, denoted as INTERCEPT*. We include β-γ^* to still account for the way the intercepts are gated by the input sentence. The initial hidden and cell state are added to β now as well, as we consider these states to be part of the model bias. Finally, to determine the dependence of the input on the gate intercepts we use an interaction set that never takes the interactions with any intercept into account: $\{\beta\text{-}\beta, \beta\text{-}\gamma^*\}$, denoted as $\neg$INTERCEPT.

4 Experimental setup

We use GCD to study how our LSTM model handles two different linguistic phenomena: subject-verb agreement and anaphora resolution in relation to *gender*. Next to the model of Gulordava et al. (for which we present our results), we also ran our experiments on the LM of Józefowicz et al. (2016), which arrives at similar results.

4.1 Subject-verb agreement

We consider a variant of the *number-agreement* (NA) task that was proposed by Linzen et al. (2016) to assess the syntax-sensitivity of language models. In this task, a model is evaluated based on its ability to track a long-distance subject-verb relation, which is assessed by the percentage of times that the verb-form it prefers matches the *number* of the syntactic subject. Commonly, the material in between subject and verb contains an *attractor* noun that competes with the syntactic subject, e.g. *The **keys** on the table **are***.

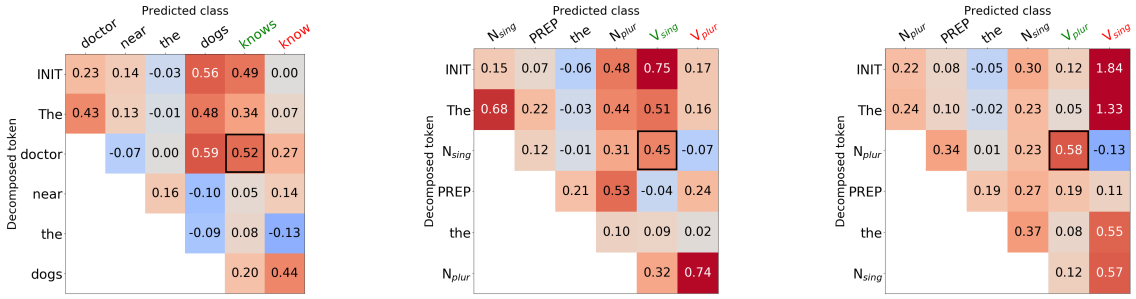
Here, we consider the NA corpora made available by Lakretz et al. (2019), which consists of a number of data sets containing a range of syntactic constructions in which number agreement plays a role. We report results for several of their data sets, but focus in particular on their *NounPP* subset, in which sentences contain an attractor embedded in a prepositional phrase. These sentences are formed following the template *The* `N Prep` *the* `N V` [..], e.g. *The boys near the car greet* [..]. The sentences in this data set are split based on the number of the subject and the attractor, resulting in four different conditions: SS, SP, PS, and PP.

4.2 Anaphora resolution and gender bias

Our second experiment concerns anaphora resolution and the possible gender biases that networks may use to perform this task. We focus on intra-sentential anaphora resolution, in which a pronoun in a subordinate clause refers to an entity in the main clause, based on *gender information*. For example: *The monk liked the **nun**, because **she** was always nice to him*.

Compared to number agreement it is more difficult to formulate a setup for anaphora resolution in which there is a right or wrong prediction that directly reflects how the model handles the phenomenon: when predicting ***she*** in the example, it could have been equally probable to predict ***he***. Rather, to establish if a model correctly resolves the referent of a pronoun, it should be checked what the model considered to be the source of this prediction, which cannot directly be inferred from the prediction itself. GCD gives us exactly this information and is therefore an excellent tool to study anaphora resolution in language modeling.

To create our corpus, we use the templates from the WinoBias corpus created by Zhao et al. (2018). This corpus contains sentences with job titles that are gender neutral, yet contain a stereotypical bias towards one gender (doctors and CEOs are *male*, nurses and housekeepers *female*). We construct two types of corpora, one containing the stereotypical job titles of Zhao et al. and one in which we replace these titles by entity descriptions that are unambiguously gendered (*king*, *bride*, *father*, etc.). Similar to the *NounPP* corpus for NA, we



(a) A single *NounPP* sentence: singular subject with plural attractor (SP).

(b) Average *NounPP* SP: singular subject with plural attractor.

(c) Average *NounPP* PS: plural subject with singular attractor.

Figure 2: Average contributions for the NounPP corpus of Lakretz et al. (2019), defined as β_t^z / z_t. INIT denotes the contribution of the initial states. The picture depicts an asymmetry in the way that the model encodes singularity and plurality: while plural verbs depend strongly on the subject, for singular sentences this is not the case.

create 4 different conditions, based on the gender of the subject and object (FF, FM, MF, and MM). An example of an MF sentence would be *The father likes the woman, because he/she*. We sample from the set of entity descriptions to create 500 sentences per condition, for both corpus types.

4.3 Experiment types

Phrase contributions In the first type of experiment, we consider the contributions of different words in the input to a later prediction of the model. This allows us to compare the contributions of different words in the sentence and track which words the model uses to come to its prediction. We compute a phrase's contribution to a prediction at step t as β_t^z / z_t.

Pruning information In the second type of experiment, we focus on the model's *predictions*. In particular, we study how the model's predictions change when it is forced to consider only specific parts of the input, by disregarding all information that does not belong to the inside information of that part of the input. This allows us to quantify the extent to which a correct prediction does in fact stem from that phrase. For this experiment, we consider several different interaction sets, that differ in what is considered to be inside the contribution of the phrase: IN describes the direct contribution of some phrase, INTERCEPT the contribution of the model intercepts, and ¬INTERCEPT the contribution of some phrase without its intercept interactions.

5 Subject-verb Agreement

We now study what information the LM uses to achieve the high prediction accuracies that were reported by Lakretz et al. (2019).

5.1 Phrase contributions

For every word in a sentence, we compute the GCD contribution for all words preceding this word. We plot these contributions in a *decomposition matrix* (akin to the attention plots often seen in machine translation papers). Every cell of this matrix represents the contribution of an input x_i (row i) to an output y_j (column j). The complete decomposition of an output word y_j can thus be found in column j. The reported scores are the decomposed scores normalised by the total model logit, resulting in the relative contribution.

In Figure 2, we plot the average decomposition matrices for the SP and PS splits of the NounPP data set. While many interesting observations can be made here, we would like to focus on the final 2 columns that represent the decompositions of the correct and wrong verb in the sentence, and on the contribution of the subject to this verb. In the singular case (2b), this contribution is, surprisingly, relatively low: The correct verb prediction does not seem to depend solely on the syntactic subject, but stems from elements that lie outside the subject as well. For the plural case, this picture is strikingly different: The highest contribution now stems from the subject of the sentence. When considering the decomposition of the wrong verb (the final column) it becomes even more clear that contributions to a plural verb predominantly stem from a plural noun, whereas singular verbs receive strong contributions from non-numbered tokens as well. This quite remarkable difference provides the first evidence for one of our conclusions: A singular prediction acts as the default number for the model, and predicting a plural verb requires

			GCD		
Task	**C**	FULL	IN	INTERCEPT*	¬INTERCEPT
Simple	S	100	73.3 (91.3)	97.3 (100)	69.7 (86.3)
Simple	P	100	100 (100)	32.7 (7.7)	100 (100)
nounPP	SS	99.2	93.0 (99.7)	99.8 (99.8)	72.7 (88.7)
nounPP	SP	87.2	90.3 (99.3)	98.8 (99.8)	60.5 (83.5)
nounPP	PS	92.0	100 (100)	0.0 (0.0)	100 (100)
nounPP	PP	99.0	100 (99.3)	7.0 (0.5)	99.8 (100)
namePP	SS	99.3	97.7 (91.3)	99.4 (100)	76.2 (90.9)
namePP	PS	68.9	98.3 (98.2)	1.3 (0.0)	99.9 (99.9)

Table 1: Accuracies on various subject-verb agreement tasks of Lakretz et al. (2019). FULL denotes the full model accuracies. IN is the decomposition of the subject, INTERCEPT* only decomposes the gate intercepts of the model. ¬INTERCEPT takes no interactions with the intercepts into account. Singular conditions are denoted in green. $(\cdot)$ denotes accuracies of scores without decoder bias, i.e. Dh_t vs $Dh_t + b_d$.

some explicit evidence coming from the subject.

5.2 Pruning information

To quantify to which extent the model bases its prediction on the subject, we prune all information that is not directly related to the subject and repeat Lakretz et al.'s NA tasks. If the model prediction were based solely on the number of the subject, its accuracy should go up, as we filter out all potentially intervening or confusing information. If, on the other hand, the prediction of the verb is not causally linked to the subject, but the model is using heuristics that require the rest of the sentence, no increase in accuracy is to be expected. We show the results, along with the accuracy of the full model in Table 1.

These numbers show a strong causal relation between plural subjects and verbs: The number prediction accuracy for the IN decomposition goes up for all cases with a plural subject. This confirms our previous finding from the decomposition matrix, which showed a relatively high contribution of plural subjects to plural verbs, as well as the conclusion of Lakretz et al. (2019) that the model is in fact keeping track of syntactic structure.

When considering the singular subjects an interesting pattern emerges: The decomposition of sentences for which the intervening attractor has the same number leads to a *lower* accuracy. This confirms that the model is in fact basing its prediction for these conditions on information that lies outside the subject itself.

Intercepts When we only decompose with respect to the gate intercepts (INTERCEPT*, column 5) it turns out the model has an extreme preference for selecting singular verbs. Decomposing without the intercept interactions (NO INTERCEPT, column 6) leads (as expected) to opposite results: the decomposed model now has a strong preference towards plural verbs as the singular prediction no longer can depend on these intercepts. This further confirms that singular verbs are used as a default baseline, which is partly encoded in its intercepts. To predict plural verbs, on the other hand, some evidence is needed, which the model picks up correctly from the subject number.

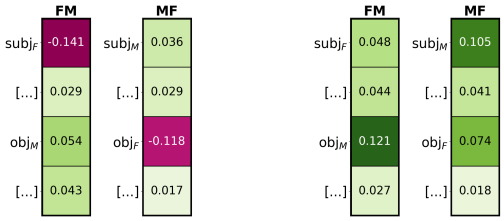
Corpus frequency One would expect that due to the model's default number being singular, this class to be more encountered during training. This turns out not to be the case: in the model's training corpus the plural verbs of the NA tasks occurred over 5 times as often as their singular counterparts. This higher frequency is in fact represented in the decoder intercept, which is higher on average for plural verbs, but it is surprising that the LSTM weights encode a default for the minority class.

6 Anaphora-resolution and gender

For the NA-tasks, the full model accuracy provides evidence that the model can perform the task well; for anaphora resolution, it is not possible to create such accuracies based on the full model predictions alone. In this section, we therefore address two different questions: 1) *Does the model correctly resolve referents?* In other words: When the model generates a male or female pronoun, does it consistently do this based on male and female referents encountered earlier in the sentence, and 2) If the model correctly performs anaphora resolution, what types of interactions and information does it use to do so? In our analysis we furthermore consider the difference between sentences with unambiguously gendered referents with sentences in which the gender of the referents is ambiguous but contains a stereotypical male or female bias.

6.1 Phrase contributions

As the template that the sentences in our anaphora data set follow is not as rigid as those of the NA tasks, creating an averaged decomposition matrix for all words in the sentences does not result in a comprehensive picture. To evaluate whether the model links pronouns to referents of the correct gender, we subtract the referent contribution to *she* from that to *he*: $\beta^z_{he}/z_{he} - \beta^z_{she}/z_{she}$. A positive



(a) unambiguous (b) stereotypical

Figure 3: Average decomposed preference of *he* over *she*, calculated as the difference between the relative contributions: $\beta^z_{he}/z_{he} - \beta^z_{she}/z_{she}$. Positive values denote male preference, negative values female preference. Phrases occurring between subject and object, and object and pronoun are denoted with `[...]`.

difference then indicates this referent had a greater contribution towards predicting *he* than *she*, and a negative difference vice versa. Little difference indicates that the referent did not contribute much to the gender of the predicted pronoun.

Unambiguous referents In Figure 3a we plot this relative contribution difference for the two conditions in our data set that contain both an unambiguous female and male referent. It is evident that the model bases its prediction on a referent of the right gender: The female subjects and objects contribute more to the prediction of *she* (reflected by the negative purple cells) and the male subjects and objects more to the prediction of *he* (the positive green cells).

Interestingly, this effect is much stronger visible for the female connections. The reason for this can be found in the model intercepts; male preference is more strongly encoded in the intercepts of the decoder: *he* has an intercept of 7.75, *she* only 6.09. This enables the model to use this male prediction as a default, similar to how singular verbs acted as a default baseline for number prediction. Akin to number agreement the model thus needs to encounter sufficient evidence of an entity being female to prefer a female pronoun. In the next section we show that this male default is encoded in the gate intercepts as well.

Stereotypical referents The intermediate conclusion that the language model performs successful anaphora resolution on our experiment also provides us the opportunity to probe the gender biases of the model. To do so, we repeat the pronoun preference test on an adapted version of the WinoBias corpus (Zhao et al., 2018), in which all referents are only stereotypically considered to be male or female (e.g., *doctor* and *nurse*). The results, plotted in Figure 3b, show that the model is very susceptible to stereotypically male referents; these decomposed scores contain an even stronger male preference than for the unambiguous corpus. The stereotypically female referents, on the other hand, do not lead to a female preference, indicating that their contribution is not considered strong enough evidence by the model to prefer a female pronoun. All the intermediate tokens exhibit a slight male preference, a pattern that is comparable to the singular bias of the NA task. From these results we conclude that the model considers a stereotypically male job occupation to be male ("*doctors* are male"), whereas this does not hold for stereotypically female jobs.

6.2 Pruning information

Following our subject-verb agreement setup, we compare the predictions of our language model when it focuses only on the subject or object of the sentence. In Table 2, we show the percentage of cases in which *he* is assigned a higher decomposed score than *she*, for both unambiguously gendered referents and stereotypically gendered referents.

FULL In the first column of Table 2a, we see that if the sentence contains referents of the same gender (MM & FF), the full model prediction almost always prefers to use a pronoun with that same gender. When both a male and female referent are present, the model has a slight preference for generating a pronoun that matches with the *subject* of the sentence (which, interestingly, is the referent that is the furthest away from the pronoun). In the stereotypical case (Table 2b), the difference between male and female sentences for the FULL scores almost disappears, showing a predominant male pronoun preference. This shows that the model by default prefers a masculine pronoun, and only when it is provided sufficient evidence of a female entity it will consider predicting *she* (similar to number agreement).

Pruning When considering the decompositions with relation to the subject or object we see that the decomposed score of a male entity in all conditions always prefers a male pronoun. For female entities this effect is slightly obscured by the male bias of the decoder intercept: The accuracies without adding this intercept highlight that female contributions lead to a strong female preference. For the stereotypical corpus this female preference is

		GCD		
	FULL	SUBJECT	OBJECT	INTERCEPT*
MM	100	100 (93.2)	100 (97.8)	100 (93.2)
MF	58.6	100 (86.4)	47.2 (0.8)	100 (96.0)
FM	37.0	29.2 (0.6)	100 (97.2)	100 (98.0)
FF	1.2	77.2 (0.8)	88.8 (1.2)	100 (92.2)

(a) *%he>she*, unambiguous referents

		GCD		
	FULL	SUBJECT	OBJECT	INTERCEPT*
MM	100	100 (100)	100 (100)	100 (88.0)
MF	94.6	100 (99.6)	95.4 (84.0)	100 (84.8)
FM	88.8	90.6 (77.4)	100 (100)	100 (91.0)
FF	84.6	92.8 (75.6)	97.4 (84.0)	100 (89.2)

(b) *%he>she*, stereotypical referents

Table 2: Gender preference on the fixed and stereotypical gender corpora. Reported scores are the percentage of times *he* is preferred over *she*. The first column denotes the gender of the subject and object. FULL denotes the full model preference, SUBJECT the decomposed score of the subject phrase (including determiners), and OBJECT the decomposed object score. INTERCEPT* is the decomposed score with relation to the intercepts only. $(\cdot)$ denotes accuracies of scores without decoder bias, i.e. Dh_t vs $Dh_t + b_d$.

far less apparent, which is in line with the results of Section 6.1. When solely considering the intercept contributions it becomes clear once more that a strong male bias is encoded in them, an effect that is further amplified by the decoder intercept.

Corpus frequency For NA the default class turned out to be less frequent in the training corpus. For our gender setup it turns out the male default is in fact the majority class, with *he* being nearly 4 times more frequent than *she*. We conclude that the default class is not directly correlated to training frequency and likely depends on the phenomenon at hand, although an investigation incorporating a wider range of models would be needed to establish this.

7 Conclusion

We propose a generalised version of Contextual Decomposition (Murdoch et al., 2018) – GCD – that allows to study specifically selected interactions of components in an LSTM language model. This enables GCD to extract the contributions of a model's intercepts, or to investigate the interactions of a phrase with other phrases and intercepts.

We analyse two linguistic phenomena in a pretrained language model: subject-verb agreement, in which *number* plays a role, and anaphora resolution for which *gender* is important. Anaphora resolution in the context of language modelling had not been investigated thoroughly before, and our setup enables this at an unprecedented level.

We trace what information the language model uses to make predictions that require gender and number information and find that, in both cases, the model applies a form of *default reasoning*, by falling back on a default class (male, singular) and predicting a female or plural token only when it is provided enough explicit evidence. As such, the decision to predict masculine and singular words can not be traced back evidently to specific information in the network inputs, but is encoded by default in the model's weights.

Our setup and results demonstrate the power of GCD, which can be applied on top of any model without additional training. Our results bear relevance for work on detecting and removing model biases, and may clarify some of the issues that were raised by Gonen and Goldberg (2019), who argue that current bias removal methods only operate on a superficial level. GCD could also be used to aid a model in guiding it towards the right flow of information, which could be applied to a wide range of applications such as the interventions of Giulianelli et al. (2018). In the future, we plan on extending GCD to other types of language models, such as the currently popular attention-based models. Furthermore, we wish to expand the capacities of GCD by improving the gate factorisation with a better Shapley value approximator, such as those proposed by Lundberg and Lee (2017) or Ancona et al. (2019). The axiomatic approach of Montavon (2019) could provide further insight into how GCD relates to other explanation methods, and we are confident that combining the strengths of GCD with that of other frameworks will ultimately lead to a more *robust* and *faithful* insight into deep neural networks.

Acknowledgements

We thank our anonymous reviewers for their useful suggestions and comments. DH and WZ are funded by the Netherlands Organization for Scientific Research (NWO), through a Gravitation Grant 024.001.006 to the Language in Interaction Consortium.

References

Marco Ancona, Cengiz Öztireli, and Markus Gross. 2019. Explaining deep neural networks with a polynomial time algorithm for shapley values approximation. In *36th International Conference on Machine Learning (ICML 2019)*.

Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. Explaining recurrent neural network predictions in sentiment analysis. *EMNLP 2017*, page 159.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the Hood: Using Diagnostic Classifiers to Investigate and Improve how Language Models Track Agreement Information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 609–614.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 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*, volume 1, pages 1195–1205.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Rafal Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *CoRR*, abs/1602.02410.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do Language Models Understand Anything? On the Ability of LSTMs to Understand Negative Polarity Items. *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231.

Jaap Jumelet and Dieuwke Hupkes. 2019. diagnnose: A neural net analysis library.

Yair Lakretz, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. In *Proceedings of NAACL-HTL 2019*, pages 11–20. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *TACL*, 4:521–535.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4765–4774.

Rebecca Marvin and Tal Linzen. 2018. Targeted Syntactic Evaluation of Language Models. In *EMNLP*, pages 1192–1202. Association for Computational Linguistics.

Grégoire Montavon. 2019. *Gradient-Based Vs. Propagation-Based Explanations: An Axiomatic Comparison*, pages 253–265. Springer International Publishing, Cham.

W. James Murdoch, Peter J. Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from lstms. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Christopher Olah. 2015. Understanding lstm networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

Nina Poerner, Benjamin Roth, and Hinrich Schütze. 2018. Evaluating neural network explanation methods using hybrid documents and morphosyntactic agreement. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 340–350, Stroudsburg, PA. Association for Computational Linguistics (ACL).

Lloyd S. Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games*, (28):307–317.

Chandan Singh, W. James Murdoch, and Bin Yu. 2019. Hierarchical interpretations for neural network predictions. In *ICLR*.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN Language Models Learn about Filler-Gap Dependencies? *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20.