Attention vs non-attention for a Shapley-based explanation method

Tom Kersten
University of Amsterdam
t.kersten@uva.nl

Hugh Mee Wong
University of Amsterdam
h.m.wong@uva.nl

Jaap Jumelet
ILLC, University of Amsterdam
j.w.d.jumelet@uva.nl

Dieuwke Hupkes
Facebook AI Research
dieuwkehupkes@fb.com

Abstract

The field of explainable AI has recently seen an explosion in the number of explanation methods for highly non-linear deep neural networks. The extent to which such methods – that are often proposed and tested in the domain of computer vision – are appropriate to address the explainability challenges in NLP is yet relatively unexplored. In this work, we consider *Contextual Decomposition* (CD) – a Shapley-based input feature attribution method that has been shown to work well for recurrent NLP models – and we test the extent to which it is useful for models that contain attention operations. To this end, we extend CD to cover the operations necessary for attention-based models. We then compare how long distance subject-verb relationships are processed by models with and without attention, considering a number of different syntactic structures in two different languages: English and Dutch. Our experiments confirm that CD can successfully be applied for attention-based models as well, providing an alternative Shapley-based attribution method for modern neural networks. In particular, using CD, we show that the English and Dutch models demonstrate similar processing behaviour, but that under the hood there are consistent differences between our attention and non-attention models.

1 Introduction

Machine learning models using deep neural architectures have seen tremendous performance improvements over the last few years. The advent of models such as LSTMs (Hochreiter and Schmidhuber, 1997) and, more recently, attention-based models such as Transformers (Vaswani et al., 2017) have allowed some language technologies to reach near human levels of performance. However, this performance has come at the cost of the interpretability of these models: high levels of non-linearity make it a near impossible task for a human to comprehend how these models operate.

Understanding how non-interpretable black box models make their predictions has become an active area of research in recent years (Hupkes et al., 2018; Jumelet and Hupkes, 2018; Samek et al., 2019; Linzen et al., 2019; Tenney et al., 2019; Ettinger, 2020, i.a.). One popular interpretability approach makes use of *feature attribution methods*, that explain a model prediction in terms of the *contributions* of the input features. For instance, a feature attribution method for a sentiment analysis task can tell the modeller how much each of the input words contributed to the decision of a particular sentence.

Multiple methods of assigning contributions to the input feature approaches exist. Some are based on local model approximations (Ribeiro et al., 2016), others on gradient-based information (Simonyan et al., 2014; Sundararajan et al., 2017) and yet others consider perturbation-based methods (Lundberg and Lee, 2017) that leverage concepts from game theory such as Shapley values (Shapley, 1953). Out of these approaches the Shapley-based attribution methods are computationally the most expensive, but they are better able at explaining more complex model dynamics involving feature interactions. This makes these methods well-suited for explaining the behaviour of current NLP models on a more linguistic level.

In this work, we therefore focus our efforts on that last category of attribution methods, focusing in particular on a method known as Contextual Decomposition (CD, Murdoch et al., 2018), which provides a polynomial approach towards approximating Shapley values. This method has been shown to work well on recurrent models without attention (Jumelet et al., 2019; Saphra and Lopez, 2020), but has not yet been used to provide insights into the linguistic capacities of attention-based models. Here, to investigate the extent to which this method is also applicable for attention

Proceedings of Deep Learning Inside Out (DeeLIO):
The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 129–139
Online, June 10, 2021. ©2021 Association for Computational Linguistics

based models, we extend the method to include the operations required to deal with attention-based models and we compare two different recurrent models: a multi-layered LSTM model (similar to Jumelet et al., 2019), and a Single Headed Attention RNN (SHA-RNN, Merity, 2019). We focus on the task of *language modelling* and aim to discover simultaneously whether attribution methods like CD are applicable when attention is used, as well as how the attention mechanism influence the resulting feature attributions, focusing in particular on whether these attributions are in line with human intuitions. Following, i.a. Jumelet et al. (2019), Lakretz et al. (2019) and Giulianelli et al. (2018), we focus on how the models process long-distance subject verb relationships across a number of different syntactic constructions. To broaden our scope, we include two different languages: English and Dutch.

Through our experiments we find that, while both English and Dutch language models produce similar results, our attention and non-attention models behave differently. These differences manifest in incorrect attributions for the subjects in sentences with a plural subject-verb pair, where we find that a higher attribution is given to a plural subject when a singular verb is used compared to a singular subject.

Our main contributions to the field thus lie in two dimensions: on the one hand, we compare attention and non-attention models with regards to their explainability. On the other hand, we perform our analysis in two languages, namely Dutch and English, to see if patterns hold in different languages.

2 Background

In this section we first discuss the model architectures that we consider. Following this, we explain the attribution method that we use to explain the different models. Finally, we consider the task which we use to extract explanations.

2.1 Model architectures

To examine the differences between attention and non-attention models, we look at one instance of each kind of model. For the attention model, we consider the Single Headed Attention RNN (SHA-RNN, Merity, 2019), and for our non-attention model a multi-layered LSTM (Gulordava et al., 2018). Since both models use an LSTM at their core, we hope to capture and isolate the influence of the attention mechanism on the behaviour of the model. Using a Transformer architecture instead would have made this comparison far more challenging, given that these kinds of models differ in multiple significant aspects from LSTMs with regards to their processing mechanism. Below, we give a brief overview of the SHA-RNN architecture.

SHA-RNN The attention model we consider is the Single Headed Attention RNN, or SHA-RNN, proposed by Merity (2019). The SHA-RNN was designed to be a reasonable alternative to the comparatively much larger Transformer models. Merity argues that while larger models can bring better performance, this often comes at the cost of training and inference time. As such, the author proposed this smaller model, which achieves results comparable to earlier Transformer models, without hyperparameter tuning.

The SHA-RNN consists of a block structure with three modules: an LSTM, a pointer-based attention layer and a feed-forward Boom layer (we provide a graphical overview in Figure 1). These blocks can be stacked to create a similar setup to that of an encoder Transformer. Layer normalisation is applied at several points in the model.

The attention layer in the SHA-RNN uses only a single attention head, creating a similar mechanism to Grave et al. (2017) and Merity et al. (2017). This is in contrast to most other Transformer (and thus attention) models, which utilise multiple attention heads. However, recent work, like Michel et al. (2019), has shown that using only a single attention head may in some cases provide similar performance to a multi-headed approach, while significantly reducing the computational cost. Importantly, when using multiple blocks of the SHA-RNN, the attention layer is only applied in the second to last block.

The Boom layer represents the feed-forward layers commonly found in Transformer models (Vaswani et al., 2017). In his work, Merity uses a single feed-forward layer with a GELU activation (Hendrycks and Gimpel, 2016), followed by summation over the output to reduce the dimension of the resulting vector to that before applying the feed-forward layer.

2.2 Contextual Decomposition

The interpretability method that we use and extend in this paper is Contextual Decomposition (CD

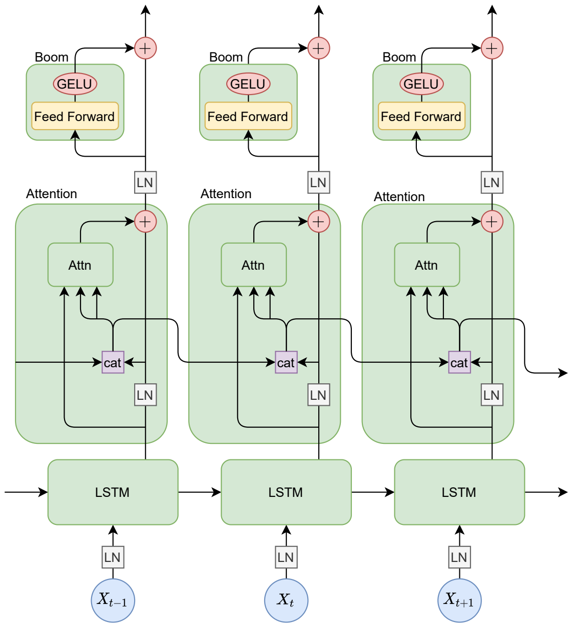


Figure 1: A schematic overview of a block in the SHA-RNN. A block in the SHA-RNN is composed of an LSTM, a single headed attention layer and a Boom feed-forward layer. Throughout the model, layer normalisation is used. Hidden states are passed between subsequent steps in the model. The memory state is concatenated with previous memory states, and passed on as well.

Murdoch et al., 2018), a feature attribution method for explaining individual predictions made by an LSTM. CD decomposes the output into a sum of two contribution types $\beta + \gamma$: one part resulting from a specific “relevant” token or phrase (β), and one part resulting from all other input to the model (γ), which is said to be “irrelevant”. The token or phrase of interest is provided as an additional parameter to the model.

CD performs a modified forward pass through the model for each individual token in the input sentence. The $\beta + \gamma$ decomposition is achieved by splitting up the hidden and cell state of the LSTM into two parts as well:

$$h_t = \beta_t + \gamma_t \tag{1}$$

$$c_t = \beta_t^c + \gamma_t^c \tag{2}$$

This decomposition is constructed such that β corresponds to contributions made solely by elements in the relevant phrase, while γ represents all other contributions. Fundamental to CD is the role of interactions between β and γ terms that arrive from operations such as (point-wise) multiplications. CD resolves this by “factorizing” the outcome of a non-linear activation function into a sum of components, based on an approximation of the Shapley value of the activation function (Shapley, 1953).

For example, the forget gate update of the cell state in an LSTM is defined as

$$c_t' = c_{t-1} \odot \sigma(W_f x_t + V_f h_{t-1} + b_f) \tag{3}$$

where $W_f \in \mathbb{R}^{d_x \times d_h}$, $V_f \in \mathbb{R}^{d_h \times d_h}$ and $b_f \in \mathbb{R}^{d_h}$. CD decomposes both c_{t-1} and h_{t-1} into a sum of β and γ terms:

$$\begin{aligned} c_t' = &\ (\beta_{t-1}^c + \gamma_{t-1}^c) \\ &\odot \sigma(W_f x_t + V_f(\beta_{t-1} + \gamma_{t-1}) + b_f) \end{aligned} \tag{4}$$

The forget gate is then decomposed into a sum of four components (x, β, γ & b_f), based on their Shapley values, which leads to a cross product between the terms in the decomposed cell state, and the decomposed forget gate. The $\beta + \gamma$ decomposition of the new cell state c_t is formed by determining which specific interactions between β and γ components should be assigned to the new β_t^c and γ_t^c terms.

In this work, we consider the generalisation of the CD method proposed by Jumelet et al. (2019), namely Generalized Contextual Decomposition (GCD). They alter the way that β and γ interactions are divided over these terms. As such, this method provides a more complete picture of the interactions within the model. For a more detailed explanation of the procedure we refer to the original papers.

2.3 Number Agreement Task

To test our models, we consider the Number Agreement (NA) task, a linguistic task that has stood central in various works in the interpretability literature (Lakretz et al., 2019; Linzen et al., 2016; Gulordava et al., 2018; Wolf, 2019; Goldberg, 2019). In this task, a model is evaluated by how well it is able to track the subject-verb relations over long distances, as assessed by the percentage of cases in which the model is able to match the form of the verb to the number of the subject. The challenge in the NA task lies in the presence of one or more attractor nouns between the subject and the verb that competes with the subject. For instance in the sentence "The boys at the car greet", "car" forms the attractor noun, and is a different number than the boys, thereby possibly confusing the model to predict a singular verb, "greets".

Several earlier studies preceded us in considering number agreement as a means to investigate language models. Linzen et al. laid the groundwork for this task, using it to assess the ability of LSTMs to learn syntax-sensitive dependencies. In their work, they only considered the English language. Gulordava et al. (2018) extended the task to the Italian, Hebrew and Russian languages. Moreover, they provided a more in-depth study of the Italian model, comparing it to human subjects. Lakretz et al. (2019) provided a detailed look at the underlying mechanisms of LSTMs by which they are able to model grammatical structure. To this end, they performed an ablation study and discovered which units were mainly responsible for this mechanism. Finally, further research into the Italian version of the NA task in Lakretz et al. (2020) investigated how emergent mechanisms in language models relate to linguistic processing in humans.

Number agreement has also been explored before in the context of attribution methods. Due to the clear dependency between a subject and a verb, it is a useful task to evaluate whether a model based its prediction of the verb on the number information of the subject. Poerner et al. (2018) provide a large suite of evaluation tasks for attribution methods including number agreement, and show that attribution methods can sometimes yield unexpected contribution patterns. Jumelet et al. (2019) employ Contextual Decomposition to investigate the behaviour of an LSTM LM on a number agreement task, and demonstrate that their model employs a *default reasoning* heuristic when resolving the task, with a strong bias for singular verbs. Hao (2020) investigates an attribution method on a range of number agreement constructions containing relative clauses, showing that LMs possess a robust notion of number information.

3 Method

In this section, we first look at extending Contextual Decomposition for the SHA-RNN. Following this, we outline the models which we will use for our experiments. Finally, we explain how we extended the Number Agreement task and how we applied Contextual Decomposition to the NA task, forming the Subject Attribution task.

3.1 Contextual Decomposition for the SHA-RNN

The original Contextual Decomposition paper (Murdoch et al., 2018) only defines the decomposition for an LSTM model. The SHA-RNN also contains several operations that have not previously been covered by these two papers. As such, we have defined the decompositions for the following two operations: Layer Normalization (Ba et al., 2016) and the Softmax operation in the Single Headed Attention layer (Merity, 2019). Based on these new decompositions, we leverage the implementation of Contextual Decomposition in the `diagNNose` library of Jumelet (2020) to also cover our SHA-RNN.

Layer Normalization Layer Normalization estimates the normalization statistics over the summed inputs to the neurons in a hidden layer. A definition of the Layer Normalization operation can be found in Eq. (5).

$$\begin{aligned}\mu &= \frac{1}{n}\sum_{i=1}^{n} a_i,\\ \sigma &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(a_i-\mu)^2},\\ \mathrm{LN}(a) &= \alpha\frac{a-\mu}{\sigma}+\delta,\end{aligned} \tag{5}$$

where a represents the inputs to the hidden layer, n the number of hidden units and α and δ are learnable parameters.

Because it looks at all inputs in a layer, both β and γ might interact within this layer. As such, we must define how we handle the decomposition of this operation, which we show in Eq. (6).

$$\begin{aligned}\beta^{l+1} &= \mathrm{LN}(\beta^l)-\delta,\\ \gamma^{l+1} &= \mathrm{LN}(\beta^l+\gamma^l)-\mathrm{LN}(\beta^l)+\delta\\ \mathrm{LN}(a) &= \mathrm{LN}(\beta^l+\gamma^l) = \beta^{l+1}+\gamma^{l+1}\end{aligned} \tag{6}$$

Our decomposition strictly separates the γ contributions from the β contributions, which means that no information from γ may be captured in β.

Softmax Similar to our treatment of the Layer Normalization operation, we strictly separate γ from the β components, as can be observed in Eq. (7).

$$\begin{aligned}\beta^{l+1} &= \mathrm{Softmax}(\beta^l),\\ \gamma^{l+1} &= \mathrm{Softmax}(\beta^l+\gamma^l)-\beta^{l+1}\end{aligned} \tag{7}$$

3.2 Models

For our experiments we consider two types of models: the attention SHA-RNN model and the non-attention LSTM model. Below, we will outline the specific architectures used and training hyperparameters chosen to build and train these models.

3.2.1 Architectures

LSTM model The LSTM model we use is similar to the one used by Gulordava et al. (2018). The model is a stacked two layer LSTM, each with 650 hidden units. Word embeddings are trained alongside the model and the weights of the embedding layer are tied to the decoder layer (Inan et al., 2017).

SHA-RNN model For our SHA-RNN we use two blocks (see Fig. 1), each with an LSTM with 650 hidden units. Furthermore, our model also utilises a trained word embedding layer with tied weights, similar to our non-attention model. Finally, our Boom layer does not increase our dimension size, but keeps it at 650. This means our Boom layer reduces to a feed-forward layer with GELU activations.

3.2.2 Training

We trained four models to conduct our experiments on. For both the attention (SHA-RNN) and non-attention (LSTM) model architectures, a model was trained on a Dutch and English corpus. Both corpora are based on wikipedia text. Following Gulordava et al. (2018), only the 50.000 most common words were retained in the vocabulary for both corpora, replacing all other words with `<unk>` tokens. The corpora were split into a training, validation and test set.

The training of the models is split up in two phases: first, the model is trained for thirty epochs with a learning rate of 0.02 and a batch size of 64. Then, we fine-tune the model for an additional five epochs with the learning rate halved to 0.01 and a batch size of 16. During training, we set dropout to 0.1. We use the LAMB optimizer (You et al., 2019) following Merity (2019).

3.3 Extending Number Agreement

In this work, we extend the Number Agreement (NA) task to the Dutch language. We do so by applying the same procedure that was used in Lakretz et al. (2019), namely by creating a synthetic dataset. This is different from the works of Linzen et al. (2016) and Gulordava et al. (2018), which derived their sentences directly from corpora.

Our version of the NA task contains a total of five different templates. First of all, we use a simple template called Simple in which the verb immediately follows the subject. We then extend this by adding a prepositional phrase which modifies the subject between the subject and the verb, either by having a prepositional phrase containing a noun (NounPP) or containing a proper noun (NamePP). We then have the sentence conjunction (SConj) task, which consists of two Simple templates separated by a conjunction. The challenge of the SConj task is correctly predicting the number of the verb in the second sentence. Finally, we have the ThatNounPP template, which contains a declarative content clause which incorporates a second subject-verb dependency with a noun modifying prepositional phrase in its that-clause. An overview of the templates including example sentences can be found in Table 1.

We create our final NA-task by obtaining frequent words from our corpus to populate these sentence templates. This process is done for both the Dutch and the English corpora, such that we can more easily compare the results.

3.4 Subject Attribution Task

We propose a new task for input feature attribution methods based on the Number Agreement task: Subject Attribution. The goal of the task to produce explanations in such a way that congruent subject-verb relations gain higher attributions than non-congruent ones.

In context of the NA task this means that we compare the attribution scores of the subject of the sentence in the case where it is and is not congruent with the number of the verb. In our evaluation we consider a higher attribution for the congruent noun compared to the non-congruent noun to be correct, as this would be in line with human intuition. A schematic overview of this task can be found in Fig. 2.

In this work, we use the task in the following way: we apply our attribution method on each sentence within our dataset, generating input feature attributions. We then compare the subject attributions of these sentences to find in which percentage of the sentences the attributions for the subject were higher for the congruent verb than the non congruent one.

NA-task	Template	Example
Simple	DET N V	De jongen groet The boy greets
NounPP	DET N PREP DET N V	De jongens bij de auto groeten The boys at the car greet
NamePP	DET N PREP NAME V	De jongens bij Pat groeten The boys at Pat greet
SConj	DET N V en/and DET N V	De jongen groet en de moeders missen The boy greets and the mothers miss
ThatNounPP	DET N V dat/that DET N PREP DET N V	De jongen denkt dat de moeders bij de auto missen The boy thinks that the mothers at the car miss

Table 1: Overview of the templates for the NA-tasks. DET is a determiner, N a noun, NAME a name of a person, V a verb and PREP a preposition. The underlined noun in the template signifies the subject belonging to the relevant verb.

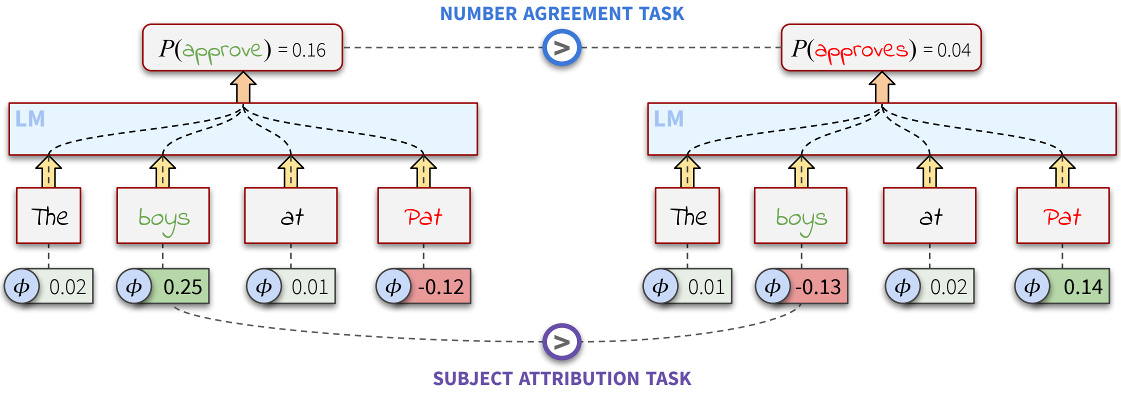


Figure 2: Schematic overview of the default **number agreement task** that compares the output probabilities of the LM, and the **subject attribution task** that compares the attribution scores of the subject to the correct and incorrect form of the verb. We hypothesise that for a model with a sophisticated understanding of number agreement, the subject's contribution to the correct verb form is greater than to the incorrect form.

4 Results and analysis

In our work, we have considered several experiments. Firstly, we evaluate the ability of our models to handle the data itself by comparing the model perplexities. Following this, we look at the Number Agreement and Subject Attribution tasks to evaluate the differences between our models.

4.1 Model Perplexities

To establish the adequacy of our models on the data, we calculate the perplexity for each model over the held-out test set (Table 2). Due to the different data sets used for the two languages, direct comparisons between the perplexity scores for the English and Dutch models are not feasible. We do observe that for both languages, the SHA-RNN yields a perplexity score that is 5% lower than the score of the LSTM counterpart.

Model	Perplexity
LSTM (English)	56.24
LSTM (Dutch)	34.24
SHA-RNN (English)	53.25
SHA-RNN (Dutch)	32.54

Table 2: Model perplexities

4.2 Number Agreement

To assess the performance of the different language models, we consider the different sentence structures presented in Table 1. For each sentence structure, we evaluate the predictive performance of the model on matching the form of the verb to the number of the relevant subject. For example, given a singular subject, we evaluate $p(\textsc{verb}_\textsc{s}|\textsc{subj}_\textsc{s}) > p(\textsc{verb}_\textsc{p}|\textsc{subj}_\textsc{s})$. The same sentence templates have been used for the Subject Attribution task. We apply Contextual Decomposition to the sentences to investigate the behavioural differences between the models.

We examine the results of our experiments along two axes: language and attention. First, we compare the Dutch and English language models. Following this, we analyse the differences between the attention and non-attention models.

4.2.1 Language axis

Across the board, the Dutch models perform slightly better on the NA tasks than the English models. This could be due to the data sets used, as the Dutch data set was larger than the English one, giving the Dutch model more opportunities to learn. We do find similar patterns between the Dutch models (Table 3a) and the English models (Table 3b): between the two languages, the models generally share the tasks and conditions that they perform well on. There are exceptions to this, as in the case of the Simple NA task for the LSTM, with Dutch models performing better on the singular condition while their English counterparts achieve higher scores on the plural condition.

When we compare the results of the models on the Subject Attribution task in Tables 3a and 3b, we find more substantial differences between the models across the languages. In case of the English models, the SHA-RNN performed rather poorly on the plural conditions of the Subject Attribution task. This is remarkable, given that the Dutch SHA-RNN yields significantly higher scores on these conditions.

We observe that for the English SHA-RNN, contextual decomposition consistently yields attribution scores that are lower for the plural conditions than those for the singular conditions (see Fig. 3 for an example). In the Dutch SHA-RNN, this behaviour is only apparent for the Simple, NounPP and NamePP tasks.

Jumelet et al. (2019) encountered similar behaviours when applying CD to an LSTM language model. They attributed the lower attributions to a bias towards singular verbs in the model, which resulted in a form of default reasoning. However, our accuracy results do not indicate a similar bias, as we found all our models performing well on both plural and singular subjects. This raises the question as to what is causing this behaviour, which we leave for future work.

Overall, these results do not demonstrate any significant differences between the Dutch and English models. While we have shown that differences occur across conditions, we find that for most conditions, both models behave similarly, with the two LSTM models displaying more similarities than the SHA-RNN models.

4.2.2 Attention axis

To compare the attention models (SHA-RNNs) to the non-attention model (the LSTMs), we again first consider the accuracy scores in Tables 3a and 3b. A comparison between the SHA-RNN and the LSTM shows that the SHA-RNN performs slightly worse than the LSTM by a small margin. There are some cases where this difference is more pronounced, such as for the English ThatNounPP task (see Table 3b), where we observe large differences for the singular subject conditions. This behaviour goes against the perplexity results in Table 2, which indicate a better performing SHA-RNN. This is in line with the results found by Nikoulina et al. (2021), who demonstrate that perplexity is not always directly correlated to performance on downstream tasks, as appears to be the case for our Number Agreement task.

Looking at the model explanations in Tables 3a and 3b we see that across the board the LSTM performs better on the Subject Attribution task. We find that both SHA-RNN models generally do not produce the expected attributions for the plural subject conditions, while there are very few instances of the LSTM performing under 50%, only failing by a large margin for the English LSTM on the Simple P and NamePP P conditions (see Table 3a).

From our observations, the attention and non-attention models behave differently both in terms of accuracy scores on the NA task and the explanations from the Subject Attribution task. We find that the difference between the architectures of the SHA-RNN and the LSTM leads to significant variations in general performance as well as behavioural patterns.

5 Conclusion

In this paper, we compared both attention (SHA-RNN) and non-attention (LSTM) language models across two languages, namely Dutch and English. To test these models, we extended the Number Agreement task from Lakretz et al. (2019) to the Dutch language, which allows us to compare these models across both languages. In addition to this, we extended a feature attribution method called Contextual Decomposition (Murdoch et al., 2018) to the SHA-RNN model. We applied Contextual

NA-task	Singular Subject			Plural Subject		
	Condition	SHA-RNN	LSTM	Condition	SHA-RNN	LSTM
Simple	$\underline{S}$	92.1 (77.8)	99.2 (65.4)	$\underline{P}$	94.0 (25.9)	94.4 (58.6)
NounPP	$\underline{S}S$	99.0 (83.3)	94.7 (56.1)	$\underline{P}S$	91.5 (20.7)	98.5 (70.0)
NounPP	$\underline{S}P$	95.2 (82.0)	94.7 (48.3)	$\underline{P}P$	96.8 (21.3)	98.7 (71.2)
NamePP	$\underline{S}$	59.3 (58.3)	81.8 (57.2)	$\underline{P}$	83.8 (43.3)	75.3 (48.8)
SConj	$S\underline{S}$	95.8 (77.0)	96.0 (90.3)	$S\underline{P}$	88.7 (43.8)	89.3 (63.0)
SConj	$P\underline{S}$	42.8 (67.0)	89.5 (89.3)	$P\underline{P}$	94.0 (50.2)	95.5 (42.8)
ThatNounPP	$S\underline{S}S$	98.3 (72.2)	96.7 (80.7)	$S\underline{P}S$	99.3 (61.8)	100.0 (89.3)
ThatNounPP	$S\underline{S}P$	99.0 (65.5)	94.7 (75.2)	$S\underline{P}P$	99.2 (66.2)	100.0 (91.8)
ThatNounPP	$P\underline{S}S$	97.8 (70.7)	96.8 (83.5)	$P\underline{P}S$	99.7 (62.2)	100.0 (89.3)
ThatNounPP	$P\underline{S}P$	98.2 (62.0)	91.3 (78.0)	$P\underline{P}P$	99.5 (65.8)	100.0 (91.7)

(a) Results for the **Dutch** language models.

NA-task	Singular Subject			Plural Subject		
	Condition	SHA-RNN	LSTM	Condition	SHA-RNN	LSTM
Simple	$\underline{S}$	94.0 (93.3)	92.7 (93.3)	$\underline{P}$	99.3 (11.7)	96.3 (35.7)
NounPP	$\underline{S}S$	86.0 (92.3)	78.3 (95.5)	$\underline{P}S$	82.5 (8.8)	93.3 (54.6)
NounPP	$\underline{S}P$	83.8 (93.5)	54.8 (94.0)	$\underline{P}P$	97.0 (9.0)	96.8 (59.5)
NamePP	$\underline{S}$	68.0 (89.3)	86.7 (96.5)	$\underline{P}$	66.2 (14.5)	52.3 (12.5)
SConj	$S\underline{S}$	93.8 (93.0)	94.3 (90.5)	$S\underline{P}$	99.3 (15.7)	96.3 (87.2)
SConj	$P\underline{S}$	82.3 (93.5)	94.3 (94.3)	$P\underline{P}$	99.3 (10.7)	98.8 (90.5)
ThatNounPP	$S\underline{S}S$	91.8 (100.0)	70.7 (92.0)	$S\underline{P}S$	92.3 (5.0)	95.8 (51.3)
ThatNounPP	$S\underline{S}P$	85.2 (100.0)	43.7 (94.0)	$S\underline{P}P$	98.7 (4.2)	100.0 (65.7)
ThatNounPP	$P\underline{S}S$	86.2 (99.8)	69.7 (92.3)	$P\underline{P}S$	92.0 (4.3)	97.0 (55.7)
ThatNounPP	$P\underline{S}P$	81.2 (100.0)	46.3 (92.3)	$P\underline{P}P$	98.2 (2.3)	99.5 (68.0)

(b) Results for the **English** language models.

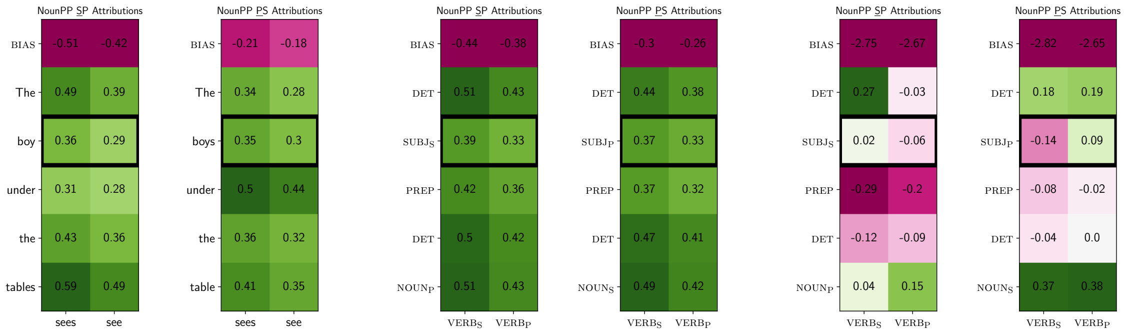
Table 3: Overview of prediction accuracy scores (the numbers outside the brackets) and subject attribution behaviour (in brackets) on the Number Agreement tasks for the Dutch and English language models. For each task, the noun inflections are given in the condition column, with S indicating singular and P indicating plural. The underlined letter in the condition indicates the noun belonging to the verb that is predicted. The numbers in brackets denote the performance on the subject attribution task: the percentage of cases in which the attributions of the subjects were higher to the congruent verb than to the non-congruent ones. The colour coding of the table cells follows the performance on this subject attribution task along a colour gradient from green (high performance) to red (low performance).

Decomposition to the Number Agreement task to obtain interpretable explanations and compared the different models from a feature attribution standpoint.

We found that both the Dutch and English models behaved similarly in terms of accuracy. While general performance differed between the two languages, we did find that similar behavioural patterns emerged from the models. This partially held for the explanations obtained through Contextual Decomposition, where we did uncover differences. These differences were centred around the SHA-RNN, which we found behaved as if it applied default reasoning similar to the work of Jumelet et al. (2019).

Comparing our attention and non-attention models, we found immediate differences, both when comparing the performance on the Number Agreement task as when looking into the attributions. Both models performed differently on the same tasks and feature attributions varied between them. We found that our LSTM performed better on the attribution task.

Our current results suggest that attention and non-attention models behave differently according to Contextual Decomposition. More specifically, we find that the attention models have more difficulty producing correct attributions for plural sentences. A logical next step would then be to compare our current results by those obtained through different attribution methods such as SHAP (Lundberg and Lee, 2017) and Integrated Gradients (Sundararajan et al., 2017). Should we find that Contextual Decomposition holds up well to these other



(a) Example SHA-RNN attributions (b) Aggregated SHA-RNN attributions (c) Aggregated LSTM attributions

Figure 3: Contextual Decomposition attributions for the English models (SHA-RNN and LSTM) on the SP and PS conditions of the NounPP task. Fig. 3a shows the attributions of two individial sentences, while Figs. 3b and 3c show aggregated attributions over all sentences of that condition. Note that in Fig. 3b the attribution for the subject under the singular verb is both higher in the SP condition as well as in PS condition, while in Fig. 3c the attribution is higher for the subject matching the verb form.

methods, it could then prove to be a valuable method for approximating Shapley values in polynomial time. Moreover, it is worth looking into the application of Contextual Decomposition in Transformer architectures, which rely more heavily on these kinds of attention mechanisms.

An alternative line of research that we would like to explore is the attention mechanism itself. Even though it has been shown that attention does not provide guarantees for explainability (Jain and Wallace, 2019), it would still be worthwhile to investigate the attention patterns that are employed by the SHA-RNN.

References

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

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, Brussels, Belgium. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing BERT's Syntactic Abilities. page 4.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2017. Improving neural language models with a continuous cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1195–1205. Association for Computational Linguistics.

Yiding Hao. 2020. Attribution analysis of grammatical dependencies in lstms. *CoRR*, abs/2005.00062.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

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

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Jaap Jumelet. 2020. diagnnose: A library for neural activation analysis. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 342–350.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of LSTMs to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Jaap Jumelet, Willem Zuidema, and Dieuwke Hupkes. 2019. Analysing neural language models: Contextual decomposition reveals default reasoning in number and gender assignment. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 1–11, Hong Kong, China. Association for Computational Linguistics.

Yair Lakretz, Dieuwke Hupkes, Alessandra Vergallito, Marco Marelli, Marco Baroni, and Stanislas Dehaene. 2020. Exploring processing of nested dependencies in neural-network language models and humans. *CoRR*, abs/2006.11098.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 11–20, Minneapolis, Minnesota. Association for Computational Linguistics.

Tal Linzen, Grzegorz Chrupała, Yonatan Belinkov, and Dieuwke Hupkes, editors. 2019. *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Florence, Italy.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Trans. Assoc. Comput. Linguistics*, 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, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774.

Stephen Merity. 2019. Single headed attention RNN: stop thinking with your head. *CoRR*, abs/1911.11423.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14014–14024.

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*. OpenReview.net.

Vassilina Nikoulina, Maxat Tezekbayev, Nuradil Kozhakhmet, Madina Babazhanova, Matthias Gallé, and Zhenisbek Assylbekov. 2021. The rediscovery hypothesis: Language models need to meet linguistics. *CoRR*, abs/2103.01819.

Nina Poerner, Hinrich Schütze, and Benjamin Roth. 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.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the Demonstrations Session, NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 97–101. The Association for Computational Linguistics.

Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. 2019. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature.

Naomi Saphra and Adam Lopez. 2020. Lstms compose—and learn—bottom-up. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2797–2809.

Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Thomas Wolf. 2019. Some additional experiments extending the tech report ”Assessing BERT's Syntactic Abilities” by Yoav Goldberg. page 7.

Yang You, Jing Li, Jonathan Hseu, Xiaodan Song, James Demmel, and Cho-Jui Hsieh. 2019. Reducing BERT pre-training time from 3 days to 76 minutes. *CoRR*, abs/1904.00962.