Representation of Linguistic Form and Function in Recurrent Neural Networks

Ákos Kádár[*]
Tilburg University

Grzegorz Chrupała[*]
Tilburg University

Afra Alishahi[*]
Tilburg University

We present novel methods for analyzing the activation patterns of recurrent neural networks from a linguistic point of view and explore the types of linguistic structure they learn. As a case study, we use a standard standalone language model, and a multi-task gated recurrent network architecture consisting of two parallel pathways with shared word embeddings: The VISUAL *pathway is trained on predicting the representations of the visual scene corresponding to an input sentence, and the* TEXTUAL *pathway is trained to predict the next word in the same sentence. We propose a method for estimating the amount of contribution of individual tokens in the input to the final prediction of the networks. Using this method, we show that the* VISUAL *pathway pays selective attention to lexical categories and grammatical functions that carry semantic information, and learns to treat word types differently depending on their grammatical function and their position in the sequential structure of the sentence. In contrast, the language models are comparatively more sensitive to words with a syntactic function. Further analysis of the most informative n-gram contexts for each model shows that in comparison with the* VISUAL *pathway, the language models react more strongly to abstract contexts that represent syntactic constructions.*

1. Introduction

Recurrent neural networks (RNNs) were introduced by Elman (1990) as a connectionist architecture with the ability to model the temporal dimension. They have proved popular for modeling language data as they learn representations of words and larger linguistic units directly from the input data, without feature engineering. Variations of the RNN architecture have been applied in several NLP domains such as parsing (Vinyals et al. 2015) and machine translation (Bahdanau, Cho, and Bengio 2015), as well

[*] Tilburg Center for Cognition and Communication, Tilburg University, 5000 LE Tilburg, The Netherlands, E-mail: {a.kadar, g.chrupala, a.alishahi}@uvt.nl.

Submission received: 21 July 2016; revised version received: 5 June 2017; accepted for publication: 7 July 2017.

doi:10.1162/COLI_a_00300

as in computer vision applications such as image generation (Gregor et al. 2015) and object segmentation (Visin et al. 2016). RNNs are also important components of systems integrating vision and language—for example, image (Karpathy and Fei-Fei 2015) and video captioning (Yu et al. 2015).

These networks can represent variable-length linguistic expressions by encoding them into a fixed-size low-dimensional vector. The nature and the role of the components of these representations are not directly interpretable as they are a complex, non-linear function of the input. There have recently been numerous efforts to visualize deep models such as convolutional neural networks in the domain of computer vision, but much less so for variants of RNNs and for language processing.

The present article develops novel methods for uncovering abstract linguistic knowledge encoded by the distributed representations of RNNs, with a specific focus on analyzing the hidden activation patterns rather than word embeddings and on the syntactic generalizations that models learn to capture. In the current work we apply our methods to a specific architecture trained on specific tasks, but also provide pointers about how to generalize the proposed analysis to other settings.

As our case study we picked the IMAGINET model introduced by Chrupała, Kádár, and Alishahi (2015). It is a multi-task, multi-modal architecture consisting of two gated-recurrent unit (GRU) (Cho et al. 2014; Chung et al. 2014) pathways and a shared word embedding matrix. One of the GRUs (VISUAL) is trained to predict image vectors given image descriptions, and the other pathway (TEXTUAL) is a language model, trained to sequentially predict each word in the descriptions. This particular architecture allows a comparative analysis of the hidden activation patterns between networks trained on two different tasks, while keeping the training data and the word embeddings fixed. Recurrent neural language models akin to TEXTUAL, which are trained to predict the next symbol in a sequence, are relatively well understood, and there have been some attempts to analyze their internal states (Elman 1991; Karpathy, Johnson, and Li 2016, among others). In constrast, VISUAL maps a complete sequence of words to a representation of a corresponding visual scene and is a less commonly encountered, but more interesting, model from the point of view of representing meaning conveyed via linguistic structure. For comparison, we also consider a standard standalone language model.

We report a thorough quantitative analysis to provide a linguistic interpretation of the networks' activation patterns. We present a series of experiments using a novel method we call **omission score** to measure the importance of input tokens to the final prediction of models that compute distributed representations of sentences. Furthermore, we introduce a more global measure for estimating the informativeness of various types of *n*-gram contexts for each model. These techniques can be applied to various RNN architectures such as recursive neural networks and convolutional neural networks.

Our experiments show that the VISUAL pathway in general pays special attention to syntactic categories that carry semantic content, and particularly to nouns. More surprisingly, this pathway also learns to treat word types differently depending on their grammatical function and their position in the sequential structure of the sentence. In contrast, the TEXTUAL pathway and the standalone language model are especially sensitive to the local syntactic characteristics of the input sentences. Further analysis of the most informative *n*-gram contexts for each model shows that whereas the VISUAL pathway is mostly sensitive to lexical (i.e., token *n*-gram) contexts, the language models react more strongly to abstract contexts (i.e., dependency relation *n*-grams) that represent syntactic constructions.

2. Related Work

The direct predecessors of modern architectures were first proposed in the seminal paper by Elman (1990). He modifies the RNN architecture of Jordan (1986) by changing the output-to-memory feedback connections to hidden-to-memory recurrence, enabling Elman networks to represent arbitrary dynamic systems. Elman (1991) trains an RNN on a small synthetic sentence data set and analyzes the activation patterns of the hidden layer. His analysis shows that these distributed representations encode lexical categories, grammatical relations, and hierarchical constituent structures. Giles et al. (1991) train RNNs similar to Elman networks on strings generated by small deterministic regular grammars with the objective to recognize grammatical and reject ungrammatical strings, and develop the **dynamic state partitioning** technique to extract the learned grammar from the networks in the form of deterministic finite state automatons.

More closely related is the recent work of Li et al. (2016a), who develop techniques for a deeper understanding of the activation patterns of RNNs, but focus on models with modern architectures trained on large scale data sets. More specifically, they train long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber 1997) for phrase-level sentiment analysis and present novel methods to explore the inner workings of RNNs. They measure the salience of tokens in sentences by taking the first-order derivatives of the loss with respect to the word embeddings and provide evidence that LSTMs can learn to attend to important tokens in sentences. Furthermore, they plot the activation values of hidden units through time using heat maps and visualize local semantic compositionality in RNNs. In comparison, the present work goes beyond the importance of single words and focuses more on exploring structure learning in RNNs, as well as on developing methods for a comparative analysis between RNNs that are focused on different modalities (language vs. vision).

Adding an explicit attention mechanism that allows the RNNs to focus on different parts of the input was recently introduced by Bahdanau, Cho, and Bengio (2015) in the context of extending the sequence-to-sequence RNN architecture for neural machine translation. On the decoding side this neural module assigns weights to the hidden states of the decoder, which allows the decoder to selectively pay varying degrees of attention to different phrases in the source sentence at different decoding time-steps. They also provide qualitative analysis by visualizing the attention weights and exploring the importance of the source encodings at various decoding steps. Similarly Rocktäschel et al. (2016) use an attentive neural network architecture to perform natural language inference and visualize which parts of the hypotheses and premises the model pays attention to when deciding on the entailment relationship. Conversely, the present work focuses on RNNs without an explicit attention mechanism.

Karpathy, Johnson, and Li (2016) also take up the challenge of rendering RNN activation patterns understandable, but use character level language models and rather than taking a linguistic point of view, focus on error analysis and training dynamics of LSTMs and GRUs. They show that certain dimensions in the RNN hidden activation vectors have specific and interpretable functions. Similarly, Li et al. (2016b) use a convolutional neural network (CNN) based on the architecture of Krizhevsky, Sutskever, and Hinton (2012), and train it on the ImageNet data set using different random initializations. For each layer in all networks they store the activation values produced on the validation set of the ImageNet Large Scale Visual Recognition Competition and align similar neurons of different networks. They conclude that although some features are learned across networks, some seem to depend on the initialization. Other works on visualizing the role of individual hidden units in deep models for vision synthesize images by optimizing random images through backpropagation to maximize the activity of units (Erhan et al. 2009; Simonyan, Vedaldi, and Zisserman 2014; Yosinski et al. 2015; Nguyen, Yosinski, and Clune 2016) or to approximate the activation vectors of particular layers (Dosovitskiy and Brox 2015; Mahendran and Vedaldi 2016).

While this paper was under review, a number of articles appeared that also investigate linguistic representations in LSTM architectures. In an approach similar to ours, Li, Monroe, and Jurafsky (2016) study the contribution of individual input tokens as well as hidden units and word embedding dimensions by erasing them from the representation and analyzing how this affects the model. They focus on text-only tasks and do not take other modalities such as visual input into account. Adi et al. (2017) take an alternative approach by introducing prediction tasks to analyze information encoded in sentence embeddings about sentence length, sentence content, and word order. Finally, Linzen, Dupoux, and Goldberg (2016) examine the acquisition of long-distance dependencies through the study of number agreement in different variations of an LSTM model with different objectives (number prediction, grammaticality judgment, and language modeling). Their results show that such dependencies can be captured with very high accuracy when the model receives a strong supervision signal (i.e., whether the subject is plural or singular), but simple language models still capture the majority of test cases. Whereas they focus on an in-depth analysis of a single phenomenon, in our work we are interested in methods that make it possible to uncover a broad variety of patterns of behavior in RNNs.

In general, there has been a growing interest within computer vision in understanding deep models, with a number of papers dedicated to visualizing learned CNN filters and pixel saliencies (Simonyan, Vedaldi, and Zisserman 2014; Mahendran and Vedaldi 2015; Yosinski et al. 2015). These techniques have also led to improvements in model performance (Eigen et al. 2014) and transferability of features (Zhou et al. 2015). To date there has been much less work on such issues within computational linguistics. We aim to fill this gap by adapting existing methods as well as developing novel techniques to explore the linguistic structure learned by recurrent networks.

3. Models

In our analyses of the acquired linguist knowledge, we apply our methods to the following models:

- IMAGINET: A multi-modal GRU network consisting of two pathways, VISUAL and TEXTUAL, coupled via word embeddings.
- LM: A (unimodal) language model consisting of a GRU network.
- SUM: A network with the same objective as the VISUAL pathway of IMAGINET, but that uses sum of word embeddings instead of a GRU.

The rest of this section gives a detailed description of these models.

3.1 Gated Recurrent Neural Networks

One of the main difficulties for training traditional Elman networks arises from the fact that they overwrite their hidden states at every time step with a new value computed from the current input $\mathbf{x_t}$ and the previous hidden state $\mathbf{h_{t-1}}$. Similarly to LSTMs, GRU networks introduce a mechanism that facilitates the retention of information over multiple time steps. Specifically, the GRU computes the hidden state at current time step $\mathbf{h}_t$, as the linear combination of previous activation $\mathbf{h_{t-1}}$, and a new *candidate* activation $\tilde{\mathbf{h}}_t$:

$$\mathrm{GRU}(\mathbf{h}_{t-1}, \mathbf{x}_t) = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \tag{1}$$

where $\odot$ is elementwise multiplication, and the update gate activation $\mathbf{z_t}$ determines the amount of new information mixed in the current state:

$$\mathbf{z}_t = \sigma_s(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1}) \tag{2}$$

The candidate activation is computed as:

$$\tilde{\mathbf{h}}_t = \sigma(\mathbf{W}\mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1})) \tag{3}$$

The reset gate $\mathbf{r_t}$ determines how much of the current input $\mathbf{x_t}$ is mixed in the previous state $\mathbf{h}_{t-1}$ to form the candidate activation:

$$\mathbf{r}_t = \sigma_s(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1}) \tag{4}$$

where $\mathbf{W}$, $\mathbf{U}$, $\mathbf{W}_z$, $\mathbf{U}_z$, $\mathbf{W}_r$ and $\mathbf{U}_r$ are learnable parameters.

3.2 Imaginet

IMAGINET, introduced in Chrupała, Kádár, and Alishahi (2015), is a multi-modal GRU network architecture that learns visually grounded meaning representations from textual and visual input. It acquires linguistic knowledge through language comprehension, by receiving a description of a scene and trying to visualize it through predicting a visual representation for the textual description, while concurrently predicting the next word in the sequence.

Figure 1 shows the structure of IMAGINET. As can be seen from the figure, the model consists of two GRU pathways, TEXTUAL and VISUAL, with a shared word embedding matrix. The inputs to the model are pairs of image descriptions and their corresponding images. The TEXTUAL pathway predicts the next word at each position in the sequence of words in each caption, whereas the VISUAL pathway predicts a visual representation of the image that depicts the scene described by the caption after the final word is received.

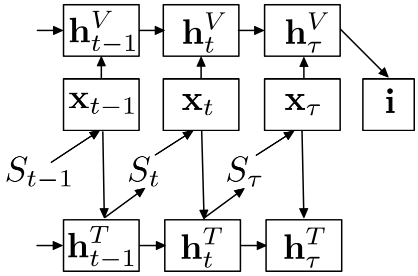

Figure 1
Structure of IMAGINET, adapted from Chrupała, Kádár, and Alishahi (2015).

Formally, each sentence is mapped to two sequences of hidden states, one by VISUAL and the other by TEXTUAL:

$$\mathbf{h}_t^V = \mathrm{GRU}^V(\mathbf{h}_{t-1}^V, \mathbf{x}_t) \tag{5}$$

$$\mathbf{h}_t^T = \mathrm{GRU}^T(\mathbf{h}_{t-1}^T, \mathbf{x}_t) \tag{6}$$

At each time step TEXTUAL predicts the next word in the sentence S from its current hidden state $\mathbf{h}_t^T$, and VISUAL predicts the image-vector[1] $\hat{\mathbf{i}}$ from its last hidden representation $\mathbf{h}_t^V$.

$$\hat{\mathbf{i}} = \mathbf{V}\mathbf{h}_\tau^V \tag{7}$$

$$p(S_{t+1}|S_{1:t}) = \mathrm{softmax}(\mathbf{L}\mathbf{h}_t^T) \tag{8}$$

The loss function is a multi-task objective that penalizes error on the visual and the textual targets simultaneously. The objective combines cross-entropy loss L^T for the word predictions and cosine distance L^V for the image predictions,[2] weighting them with the parameter α (set to 0.1).

$$L^T(\theta) = -\frac{1}{\tau}\sum_{t=1}^{\tau} \log p(S_t|S_{1:t}) \tag{9}$$

$$L^V(\theta) = 1 - \frac{\hat{\mathbf{i}} \cdot \mathbf{i}}{||\hat{\mathbf{i}}||\,||\mathbf{i}||} \tag{10}$$

$$L = \alpha L^T + (1-\alpha)L^V \tag{11}$$

For more details about the IMAGINET model and its performance, see Chrupała, Kádár, and Alishahi (2015). Note that we introduce a small change in the image representation: We observe that using standardized image vectors, where each dimension is transformed by subtracting the mean and dividing by standard deviation, improves performance.

3.3 Unimodal Language Model

The model LM is a language model analogous to the TEXTUAL pathway of IMAGINET with the difference that its word embeddings are not shared, and its loss function is the cross-entropy on word prediction. Using this model we remove the visual objective as a factor, as the model does not use the images corresponding to captions in any way.

3.4 Sum of Word Embeddings

The model SUM is a stripped-down version of the VISUAL pathway, which does not share word embeddings, only uses the cosine loss function, and replaces the GRU network with a summation over word embeddings. This removes the effect of word

[1] Representing the full image, extracted from the pre-trained CNN of Simonyan and Zisserman (2015).

[2] Note that the original formulation in Chrupała, Kádár, and Alishahi (2015) uses mean squared error instead; as the performance of VISUAL is measured on image-retrieval (which is based on cosine distances) we use cosine distance as the visual loss here.

order from consideration. We use this model as a baseline in the sections that focus on language structure.

4. Experiments

In this section, we report a series of experiments in which we explore the kinds of linguistic regularities the networks learn from word-level input. In Section 4.1 we introduce **omission score**, a metric to measure the contribution of each token to the prediction of the networks, and in Section 4.2 we analyze how omission scores are distributed over dependency relations and part-of-speech categories. In Section 4.3 we investigate the extent to which the importance of words for the different networks depends on the words themselves, their sequential position, and their grammatical function in the sentences. Finally, in Section 4.4 we systematically compare the types of *n*-gram contexts that trigger individual dimensions in the hidden layers of the networks, and discuss their level of abstractness.

In all these experiments we report our findings based on the IMAGINET model, and whenever appropriate compare it with our two other models LM and SUM. For all the experiments, we trained the models on the training portion of the MSCOCO image-caption data set (Lin et al. 2014), and analyzed the representations of the sentences in the validation set corresponding to 5000 randomly chosen images. The target image representations were extracted from the pre-softmax layer of the 16-layer CNN of Simonyan and Zisserman (2015).

4.1 Computing Omission Scores

We propose a novel technique for interpreting the activation patterns of neural networks trained on language tasks from a linguistic point of view, and focus on the high-level understanding of what parts of the input sentence the networks pay most attention to. Furthermore, we investigate whether the networks learn to assign different amounts of importance to tokens, depending on their position and grammatical function in the sentences.

In all the models the full sentences are represented by the activation vector at the end-of-sentence symbol ($\mathbf{h}_{\text{end}}$). We measure the salience of each word S_i in an input sentence $S_{1:n}$ based on how much the representation of the partial sentence $S_{\setminus i} = S_{1:i-1}S_{i+1:n}$, with the omitted word S_i, deviates from that of the original sentence representation. For example, the distance between $\mathbf{h}_{\text{end}}$(*the black dog is running*) and $\mathbf{h}_{\text{end}}$(*the dog is running*) determines the importance of *black* in the first sentence. We introduce the measure omission(i, S) for estimating the salience of a word S_i:

$$\text{omission}(i, S) = 1 - \text{cosine}(\mathbf{h}_{\text{end}}(S), \mathbf{h}_{\text{end}}(S_{\setminus i})) \tag{12}$$

Figure 2 demonstrates the omission scores for the LM, VISUAL, and TEXTUAL models for an example caption. Figure 3 shows the images retrieved by VISUAL for the full caption and for the one with the word *baby* omitted. The images are retrieved from the validation set of MSCOCO by: 1) computing the image representation of the given sentence with VISUAL; 2) extracting the CNN features for the images from the set; and 3) finding the image that minimizes the cosine distance to the query. The omission scores for VISUAL show that the model paid attention mostly to *baby* and *bed* and slightly to *laptop*, and retrieved an image depicting a baby sitting on a bed with a laptop. Removing the word *baby* leads to an image that depicts an adult male lying on a

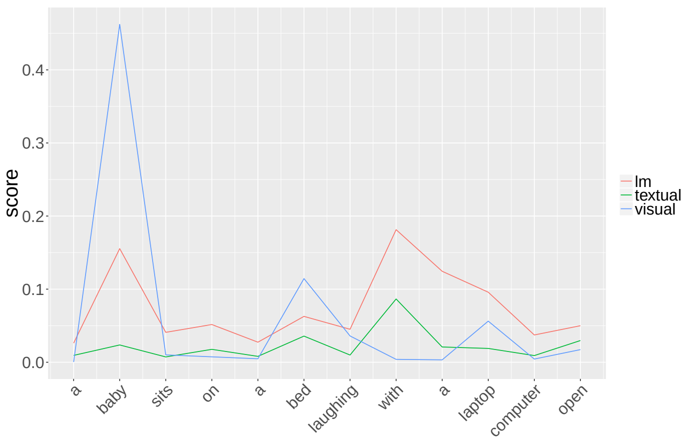

Figure 2
Omission scores for the example sentence *a baby sits on a bed laughing with a laptop computer open* for LM and the two pathways, TEXTUAL and VISUAL, of IMAGINET.

bed. Figure 2 also shows that in contrast to VISUAL, TEXTUAL distributes its attention more evenly across time steps instead of focusing on the types of words related to the corresponding visual scene. The peaks for LM are the same as for TEXTUAL, but the variance of the omission scores is higher, suggesting that the unimodal language model is more sensitive overall to input perturbations than TEXTUAL.

4.2 Omission Score Distributions

The omission scores can be used not only to estimate the importance of individual words, but also of syntactic categories. We estimate the salience of each syntactic category by accumulating the omission scores for all words in that category. We tag every word in a sentence with the part-of-speech (POS) category and the dependency relation label of its incoming arc. For example, for the sentence *the black dog*, we get

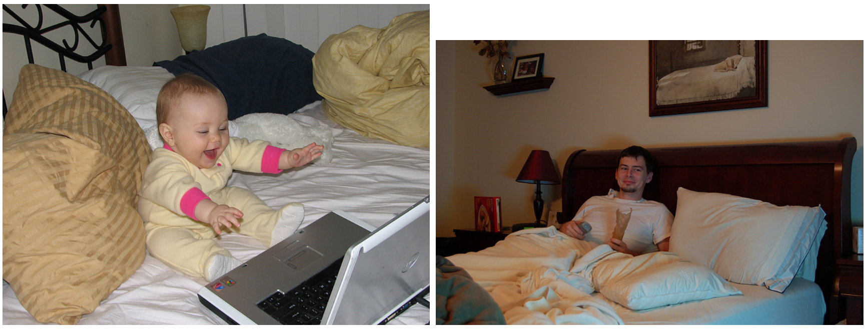

Figure 3
Images retrieved for the example sentence *a baby sits on a bed laughing with a laptop computer open* (left) and the same sentence with the second word omitted (right).

(*the*, DT, det), (*black*, JJ, amod), (*dog*, NN, root). Both POS tagging and dependency parsing are performed using the `en_core_web_md` dependency parser from the Spacy package.[3]

Figure 4 shows the distribution of omission scores per POS and dependency label for the two pathways of IMAGINET and for LM.[4] The general trend is that for the VISUAL pathway, the omission scores are high for a small subset of labels—corresponding mostly to nouns, less so for adjectives and even less for verbs—and low for the rest (mostly function words and various types of verbs). For TEXTUAL the differences are smaller, and the pathway seems to be sensitive to the omission of most types of words. For LM the distribution over categories is also relatively uniform, but the omission scores are higher overall than for TEXTUAL.

Figure 5 compares the two pathways of IMAGINET directly using the log of the ratio of the VISUAL to TEXTUAL omission scores, and plots the distribution of this ratio for different POS and dependency labels. Log ratios above zero indicate stronger association with the VISUAL pathway and below zero with the TEXTUAL pathway. We see that in relative terms, VISUAL is more sensitive to adjectives (JJ), nouns (NNS, NN), numerals (CD), and participles (VBN), and TEXTUAL is more sensitive to determiners (DT), pronouns (PRP), prepositions (IN), and finite verbs (VBZ, VBP).

This picture is complemented by the analysis of the relative importance of dependency relations: VISUAL pays most attention to the relations AMOD, NSUBJ, ROOT, COMPOUND, DOBJ, and NUMMOD, whereas TEXTUAL is more sensitive to DET, PREP, AUX, CC, POSS, ADVMOD, PRT, and RELCL. As expected, VISUAL is more focused on grammatical functions typically filled by semantically contentful words, whereas TEXTUAL distributes its attention more uniformly and attends relatively more to purely grammatical functions.

It is worth noting, however, the relatively low omission scores for verbs in the case of VISUAL. One might expect that the task of image prediction from descriptions requires general language understanding and thus high omission scores for all content words in general; however, the results suggest that this setting is not optimal for learning useful representations of verbs, which possibly leads to representations that are too task-specific and not transferable across tasks.

Figure 6 shows a similar analysis contrasting LM with the TEXTUAL pathway of IMAGINET. The first observation is that the range of values of the log ratios is narrow, indicating that the differences between these two networks regarding which grammatical categories they are sensitive to is less pronounced than when comparing VISUAL with TEXTUAL. Although the size of the effect is weak, there also seems to be a tendency for the TEXTUAL model to pay relatively more attention to content and less to function words compared with LM: It may be that the VISUAL pathway pulls TEXTUAL in this direction by sharing word embeddings with it.

Most of our findings up to this point conform reasonably well to prior expectations about effects that particular learning objectives should have. This fact serves to validate our methods. In the next section we go on to investigate less straightforward patterns.

[3] Available at `https://spacy.io/`.

[4] The boxplots in this and subsequent figures are Tukey boxplots and should be interpreted as follows: The box extends from the 25th to the 75th percentile of the data; the line across the box is the 50th percentile, and the whiskers extend past the lower and upper quartile to 1.5× the interquartile range (i.e., 75th percentile − 25th percentile); the points are outliers.

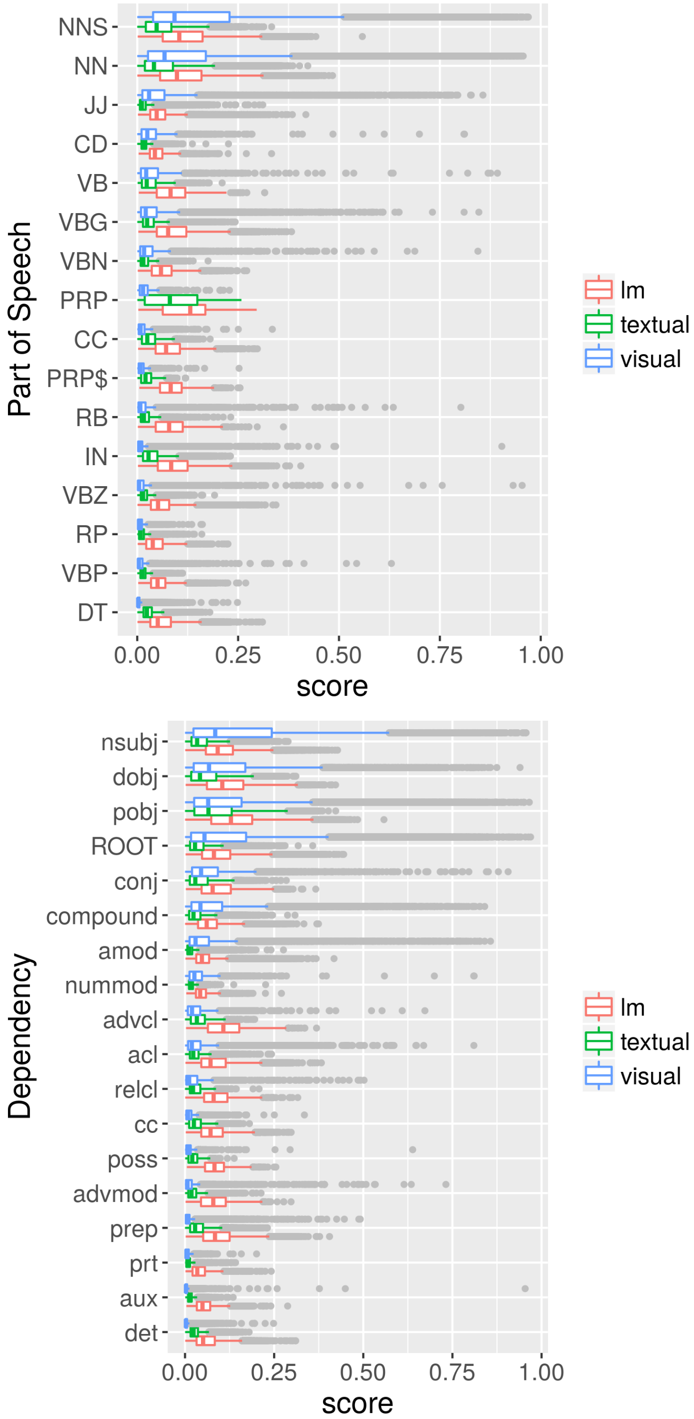

Figure 4
Distribution of omission scores for POS (left) and dependency labels (right), for the TEXTUAL and VISUAL pathways and for LM. Only labels that occur at least 1,250 times are included.

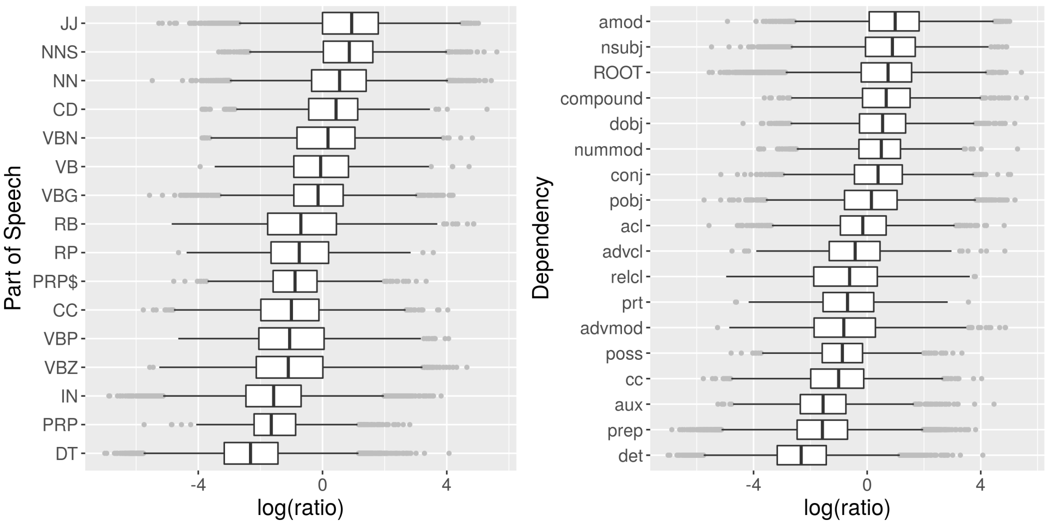

Figure 5
Distributions of log ratios of omission scores of TEXTUAL to VISUAL per POS (left) and dependency labels (right). Only labels that occur at least 1,250 times are included.

4.3 Beyond Lexical Cues

Models that utilize the sequential structure of language have the capacity to interpret the same word type differently depending on the context. The omission score distributions in Section 4.2 show that in the case of IMAGINET the pathways are differentially sensitive to content vs. function words. In principle, this may be either just due to purely lexical features or the model may actually learn to pay more attention to the same word type in appropriate contexts. This section investigates to what extent our

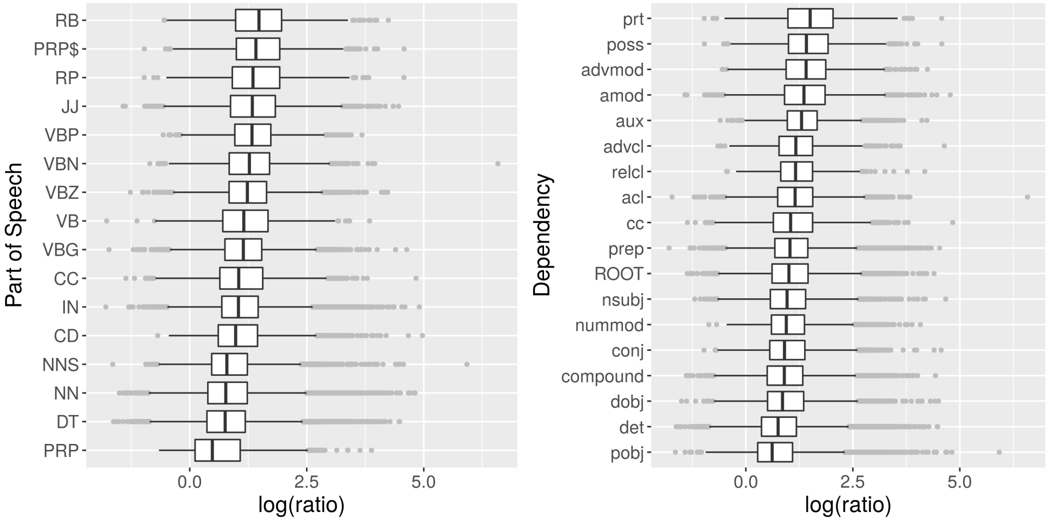

Figure 6
Distributions of log ratios of omission scores of LM to TEXTUAL per POS (left) and dependency labels (right). Only labels that occur at least 1,250 times are included.

models discriminate between occurrences of a given word in different positions and grammatical functions.

We fit four L2-penalized linear regression models that predict the omission scores per token with the following predictor variables:

1. LR WORD: word type
2. LR +DEP: word type, dependency label and their interaction
3. LR +POS: word type, position (binned as FIRST, SECOND, THIRD, MIDDLE, ANTEPENULT, PENULT, LAST) and their interaction
4. LR FULL: word type, dependency label, position, word:dependency interaction, word:position interaction

We use the 5,000-image portion of MSCOCO validation data for training and test. The captions contain about 260,000 words in total, of which we use 100,000 to fit the regression models. We then use the rest of the words to compute the proportion of variance explained by the models. For comparison we also use the SUM model, which composes word embeddings via summation, and uses the same loss function as VISUAL. This model is unable to encode information about word order, and thus is a good baseline here as we investigate the sensitivity of the networks to positional and structural cues.

Table 1 shows the proportion of variance R^2 in omission scores explained by the linear regression with the different predictors. The raw R^2 scores show that for the language models LM and TEXTUAL, the word type predicts the omission-score to a much smaller degree than VISUAL. Moreover, adding information about either the position or the dependency labels increases the explained variance for all models. However, for the TEXTUAL and LM models the position of the word adds considerable amount of information. This is not surprising considering that the omission scores are measured with respect to the final activation state, and given the fact that in a language model the recent history is most important for accurate prediction.

Figure 7 offers a different view of the data, showing the increase or decrease in R^2 for the models relative to LR +POS to emphasize the importance of syntactic structure beyond the position in the sentence. Interestingly, for the VISUAL model, dependency labels are more informative than linear position, hinting at the importance of syntactic structure beyond linear order. There is a sizeable increase in R^2 between LR +POS and LR FULL in the case of VISUAL, suggesting that the omission scores for VISUAL depend on the words' grammatical function in sentences, *even after controlling for word identity and linear position.* In contrast, adding additional information on top of lexical features in the case of SUM increases the explained variance only slightly, which is most likely due to the unseen words in the held out set.

Table 1
Proportion of variance in omission scores explained by linear regression.

	word	+pos	+dep	full
SUM	0.654	0.661	0.670	0.670
LM	0.358	0.586	0.415	0.601
TEXTUAL	0.364	0.703	0.451	0.715
VISUAL	0.490	0.506	0.515	0.523

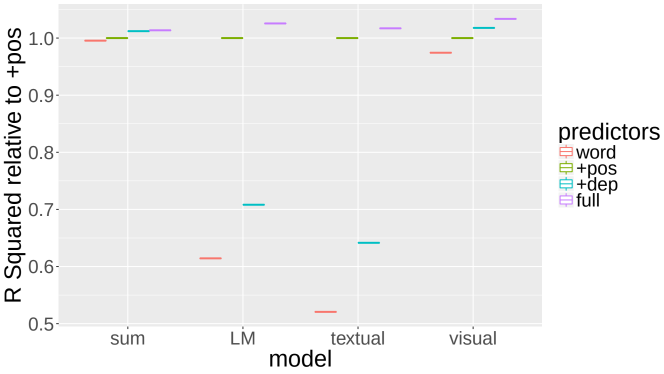

Figure 7
Proportion of variance in omission scores explained by the linear regression models for SUM, LM, VISUAL, and TEXTUAL relative to regressing on word identity and position only.

Overall, when regressing on word identities, word position, and dependency labels, the VISUAL model's omission scores are the hardest to predict of the four models. This suggests that VISUAL may be encoding additional structural features not captured by these predictors. We will look more deeply into such potential features in the following sections.

4.3.1 Sensitivity to Grammatical Function. In order to find out some of the specific syntactic configurations leading to an increase in R^2 between the LR WORD and LR +DEP predictors in the case of VISUAL, we next considered all word types with occurrence counts of at least 100 and ranked them according to how much better, on average, LR +DEP predicted their omission scores compared with LR WORD.

Figure 8 shows the per-dependency omission score distributions for seven top-ranked words. There are clear and large differences in how these words impact the

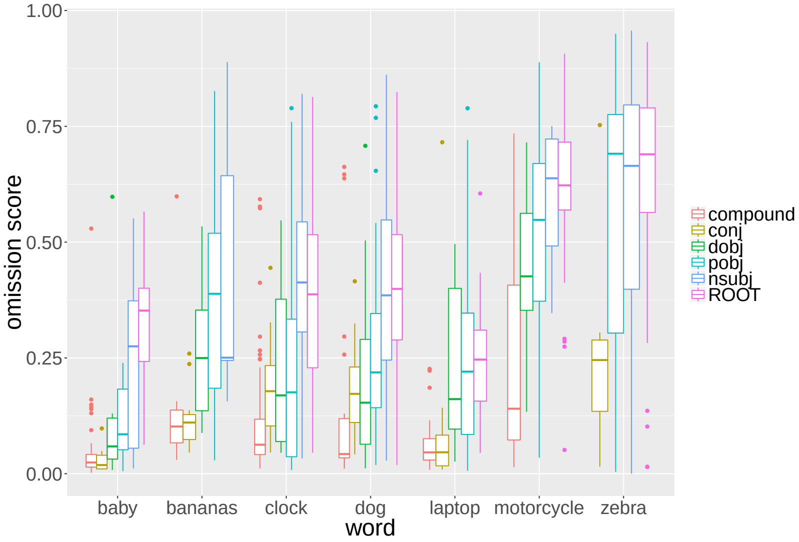

Figure 8
Distribution of omission scores per dependency label for the selected word types.

network's representation, depending on what grammatical function they fulfill. They all have large omission scores when they occur as NSUBJ (nominal subject) or ROOT, likely because these grammatical functions typically have a large contribution to the complete meaning of a sentence. Conversely, all have small omission scores when appearing as CONJ (conjunct): this is probably because in this position they share their contribution with the first, often more important, member of the conjunction—for example, in *A cow and its baby eating grass*.

4.3.2 Sensitivity to Linear Structure. As observed in Section 4.3, adding extra information about the position of words explains more of the variance in the case of VISUAL and especially TEXTUAL and LM. Figure 9 shows the coefficients corresponding to the position variables in LR FULL. Because the omission scores are measured at the end-of-sentence token, the expectation is that for TEXTUAL and LM, as language models, the words appearing closer to the end of the sentence would have a stronger effect on the omission scores. This seems to be confirmed by the plot as the coefficients for these two networks up until the *antepenult* are all negative.

For the VISUAL model it is less clear what to expect: On the one hand, because of their chain structure, RNNs are better at keeping track of short-distance rather than long-distance dependencies and thus we can expect tokens in positions closer to the end of the sentence to be more important. On the other hand, in English the information structure of a single sentence is expressed via linear ordering: The TOPIC of a sentence appears sentence-initially, and the COMMENT follows. In the context of other text types such as dialog or multi-sentence narrative structure, we would expect COMMENT to often be more important than TOPIC as COMMENT will often contain new information in these cases. In our setting of image captions, however, sentences are not part of a larger discourse; it is sentence-initial material that typically contains the most important objects depicted in the image (e.g., *two zebras are grazing in tall grass on a savannah*). Thus, for the task of predicting features of the visual scene, it would be advantageous to detect the topic of the sentence and up-weight its importance in the final meaning representation. Figure 9 appears to support this hypothesis and the network does learn to pay more attention to words appearing sentence-initially. This effect seems to be to

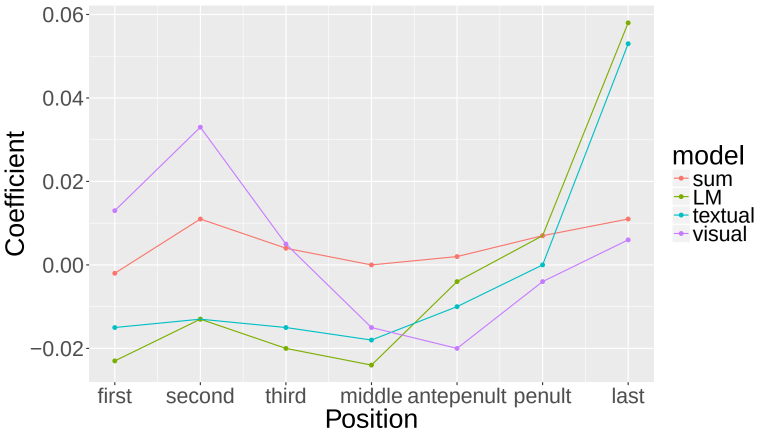

Figure 9
Coefficients on the y-axis of LR FULL corresponding to the position variables on the x-axis.

some extent mixed with the recency bias of RNNs as perhaps indicated by the relatively high coefficient of the *last* position for VISUAL.

4.4 Lexical versus Abstract Contexts

We would like to further analyze the kinds of linguistic features that the hidden dimensions of RNNs encode. Previous work (Karpathy, Johnson, and Li 2016; Li et al. 2016b) has shown that, in response to the task the networks are trained for, individual dimensions in the hidden layers of RNNs can become *specialized* in responding to certain types of triggers, including the tokens or token types at each time step, as well as the preceding context of each token in the input sentence.

Here we perform a further comparison between the models based on the hypothesis that, due to their different objectives, the activations of the dimensions of the last hidden layer of VISUAL are more characterized by semantic relations within contexts, whereas the hidden dimensions in TEXTUAL and LM are more focused on extracting syntactic patterns. In order to quantitatively test this hypothesis, we measure the strength of association between activations of hidden dimensions and either lexical (token *n*-grams) or structural (dependency label *n*-grams) types of context.

For each pathway, we define A_i as a discrete random variable corresponding to a binned activation over time steps at hidden dimension i, and C as a discrete random variable indicating the context (where C can be of type "word trigram" or "dependency label bigram," for example). The strength of association between A_i and C can be measured by their mutual information:

$$\mathrm{I}(A_i; C) = \sum_{a \in A_i} \sum_{c \in C} p(a, c) \log \left(\frac{p(a, c)}{p(a)p(c)} \right) \tag{13}$$

Similarly to Li et al. (2016b), the activation value distributions are discretized into percentile bins per dimension, such that each bin contains 5% of the marginal density. For context types, we used unigrams, bigrams, and trigrams of both dependency labels and words. Figure 10 shows the distributions of the mutual information scores for the three networks and the six context types. Note that the scores are not easily comparable between context types, because of the different support of the distributions; they are, however, comparable across the networks. The figure shows LM and TEXTUAL as being very similar, whereas VISUAL exhibits a different distribution. We next compare the models' scores pairwise to pinpoint the nature of the differences.

We use the notation MI_C^{LM}, MI_C^{T}, and MI_C^{V} to denote the median mutual information score over all dimensions of LM, TEXTUAL, and VISUAL, respectively, when considering context C. We then compute log ratios $\log(\mathrm{MI}_C^{T}/\mathrm{MI}_C^{V})$ and $\log(\mathrm{MI}_C^{LM}/\mathrm{MI}_C^{T})$ for all six context types C. In order to quantify variability we bootstrap this statistic with 5,000 replicates. Figure 11 shows the resulting bootstrap distributions for unigram, bigram, and trigram contexts, in the word and dependency conditions.

The clear pattern is that for TEXTUAL versus VISUAL, the log ratios are much higher in the case of the dependency contexts, with no overlap between the bootstrap distributions. Thus, in general, the size of the relative difference between TEXTUAL and VISUAL median mutual information score is much more pronounced for dependency context types. This suggests that features that are encoded by the hidden dimensions

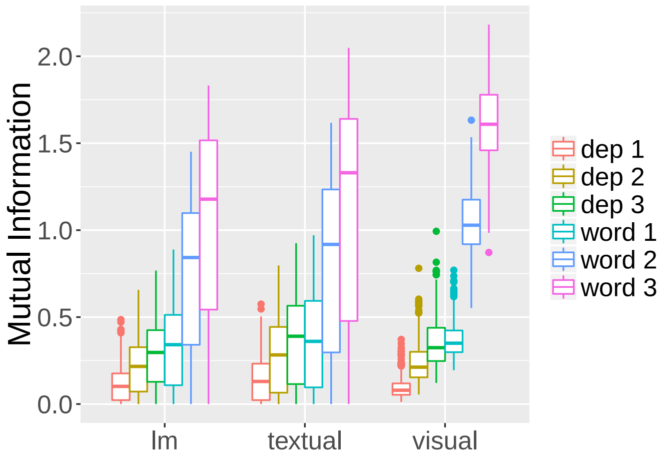

Figure 10
Distributions of the mutual information scores for the three networks and the six context types.

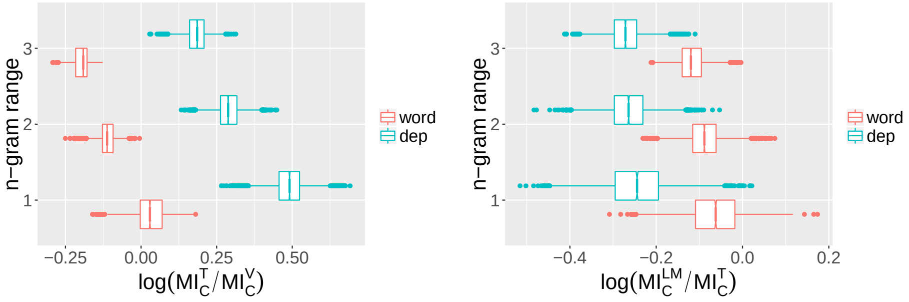

Figure 11
Bootstrap distributions of log ratios of median mutual information scores for word and dependency contexts. Left: TEXTUAL vs VISUAL; right: LM vs TEXTUAL.

of the models are indeed different, and that the features encoded by TEXTUAL are more associated with syntactic constructions than in the case of VISUAL. In contrast, when comparing LM with TEXTUAL, the difference between context types is much less pronounced, with distributions overlapping. Though the difference is small, it goes in the direction of the dimensions of the TEXTUAL model showing higher sensitivity towards dependency contexts.

The mutual information scores can be used to pinpoint specific dimensions of the hidden activation vectors that are strongly associated with a particular type of context. Table 2 lists for each network the dimension with the highest mutual information score with respect to the *dependency trigram* context type, together with the top five contexts where these dimensions carry the highest value. In spite of the quantitative difference between the networks discussed earlier, the dimensions that come up top seem to be capturing something quite similar for the three networks: (a part of) a construction with an animate root or subject modified by a participle or a prepositional phrase, though this is somewhat less clean-cut for the VISUAL pathway where only two out of five top contexts clearly conform to this pattern. Other interesting templates can be found by visual inspection of the contexts where high-scoring dimensions are active; for example,

Table 2
Dimensions most strongly associated with the dependency trigram context type, and the top five contexts in which these dimensions have high values.

Network	Dimension	Examples
LM	511	cookie/pobj attached/acl to/prep people/pobj sitting/acl in/prep purses/pobj sitting/pcomp on/prep and/cc talks/conj on/prep desserts/pobj sitting/acl next/advmod
TEXTUAL	735	male/root on/prep a/det person/nsubj rides/root a/det man/root carrying/acl a/det man/root on/prep a/det person/root on/prep a/det
VISUAL	875	man/root riding/acl a/det man/root wearing/acl a/det is/aux wearing/conj a/det a/det post/pobj next/advmod one/nummod person/nsubj is/aux

dimension 324 of LM is high for *word bigram* contexts including *people preparing, gets ready, man preparing, woman preparing, teenager preparing*.

5. Discussion

The goal of our article is to propose novel methods for the analysis of the encoding of linguistic knowledge in RNNs trained on language tasks. We focused on developing quantitative methods to measure the importance of different kinds of words for the performance of such models. Furthermore, we proposed techniques to explore what kinds of linguistic features the models learn to exploit beyond lexical cues.

Using the IMAGINET model as our case study, our analyses of the hidden activation patterns show that the VISUAL model learns an abstract representation of the information structure of a single sentence in the language, and pays selective attention to lexical categories and grammatical functions that carry semantic information. In contrast, the language model TEXTUAL is sensitive to features of a more syntactic nature. We have also shown that each network contains specialized units that are tuned to both lexical and structural patterns that are useful for the task at hand.

5.1 Generalizing to Other Architectures

For other RNN architectures such as LSTMs and their bi-directional variants, measuring the contribution of tokens to their predictions (or the omission scores) can be straightforwardly computed using their hidden state at the last time step used for prediction. Furthermore, the technique can be applied in general to other architectures that map variable-length linguistic expressions to the same fixed dimensional space and perform predictions based on these embeddings. This includes tree-structured RNN models such as the Tree-LSTM introduced in Tai, Socher, and Manning (2015), or the CNN architecture of Kim (2014) for sentence classification. However, the presented analysis and results regarding word positions can only be meaningful for RNNs as they compute their representations sequentially and are not limited by fixed window sizes.

A limitation of the generalizability of our analysis is that in the case of bi-directional architectures, the interpretation of the features extracted by the RNNs that process the input tokens in the reversed order might be hard from a linguistic point of view.

5.2 Future Directions

In the future we would like to apply the techniques introduced in this article to analyze the encoding of linguistic form and function of recurrent neural models trained on different objectives, such as neural machine translation systems (Sutskever, Vinyals, and Le 2014) or the purely distributional sentence embedding system of Kiros et al. (2015). A number of recurrent neural models rely on a so-called attention mechanism, first introduced by Bahdanau, Cho, and Bengio (2015) under the name of soft alignment. In these networks attention is explicitly represented, and it would be interesting to see how our method of discovering implicit attention, the omission score, compares. For future work we also propose to collect data where humans assess the importance of each word in a sentence and explore the relationship between omission scores for various models and human annotations. Finally, one of the benefits of understanding how linguistic form and function is represented in RNNs is that it can provide insight into how to improve systems. We plan to draw on lessons learned from our analyses in order to develop models with better general-purpose sentence representations.

References

Adi, Yossi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations (ICLR)*, Toulon, France.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representation (ICLR)*, San Diego, CA, USA.

Cho, Kyunghyun, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, pages 103–111, Doha, Qatar.

Chrupała, Grzegorz, Ákos Kádár, and Afra Alishahi. 2015. Learning language through pictures. In *Association for Computational Linguistic (ACL)*, pages 112–118, Beijing, China.

Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Deep Learning and Representation Learning Workshop*, Montreal, Quebec, Canada.

Dosovitskiy, Alexey and Thomas Brox. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196, Boston, MA.

Eigen, David, Jason Rolfe, Rob Fergus, and Yann LeCun. 2014. Understanding deep architectures using a recursive convolutional network. In *International Conference on Learning Representations (ICLR)*.

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

Elman, Jeffrey L. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7(2-3):195–225.

Erhan, Dumitru, Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2009. Visualizing higher-layer features of a deep network. In *International Conference on Machine Learning (ICML) Workshop on Learning Feature Hierarchies*, volume 1341.

Giles, C. Lee, Clifford B. Miller, Dong Chen, Guo-Zheng Sun, Hsing-Hen Chen, and Yee-Chun Lee. 1991. Extracting and learning an unknown grammar with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 317–324.

Gregor, Karol, Ivo Danihelka, Alex Graves, and Daan Wierstra. 2015. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning (ICML)*.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jordan, Michael I. 1986. Attractor dynamics and parallelism in a connectionist sequential network. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pages 531–546, Amherst, MA.

Karpathy, Andrej and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, Boston, MA.

Karpathy, Andrej, Justin Johnson, and Fei-Fei Li. 2016. Visualizing and understanding recurrent networks. In *International Conference on Learning Representations (ICLR) Workshop*, San Juan, Puerto Rico.

Kim, Yoon. 2014. Convolutional neural networks for sentence classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kiros, Ryan, Yukun Zhu, Ruslan R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Li, Jiwei, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. In *North American Chapter of the Association for Computational Linguistic (NAACL)*.

Li, Jiwei, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. arXiv preprint arXiv:1612.08220.

Li, Yixuan, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. 2016b. Convergent learning: Do different neural networks learn the same representations? In *International Conference on Learning Representation (ICLR)*.

Lin, Tsung Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft Coco: Common objects in context. In *Computer Vision–ECCV*, pages 740–755.

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

Mahendran, Aravindh and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196.

Mahendran, Aravindh and Andrea Vedaldi. 2016. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255.

Nguyen, Anh, Jason Yosinski, and Jeff Clune. 2016. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. In *Visualization for Deep Learning Workshop at International Conference on Machine Learning (ICML)*.

Rocktäschel, Tim, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)*.

Simonyan, K. and A. Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representation (ICLR) Workshop*.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Association for Computational Linguistic (ACL)*.

Vinyals, Oriol, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2755–2763.

Visin, Francesco, Kyle Kastner, Aaron Courville, Yoshua Bengio, Matteo Matteucci, and Kyunghyun Cho. 2016. ReSeg: A recurrent neural network for object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Yosinski, Jason, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding neural networks through deep visualization. In *International Conference on Machine Learning (ICML)*.

Yu, Haonan, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. 2015. Video paragraph captioning using hierarchical recurrent neural networks. In *Describing and Understanding Video & The Large Scale Movie Description Challenge (LSMDC) at International Conference on Computer Vision (ICCV)*.

Zhou, Bolei, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Object detectors emerge in deep scene CNNs. In *International Conference on Learning Representations (ICLR)*.