Evaluating the syntactic competence of RAN language models

Mario Giulianelli
11567252

Florian Mohnert
11770929

Dmitrii Krasheninnikov
11719230

Abstract

Recurrent additive network (RAN) is a new gated RNN architecture whose states are weighted sums of the linearly transformed input vectors, which allows to trace the importance of each of the previous elements of the sequence when predicting the next element if the sequence. This paper inspects a RAN language model's ability to capture syntactic information by examining its performance on the linguistic phenomena of *subject-verb agreement* and *verb form government*. We observe that the RAN tends to remember content words most strongly, and does not seem to specifically remember words that are relevant for dependency constructions. As a result, the model performs poorly on grammaticality judgements.

1 Introduction

Language modelling is the task of fitting a probability distribution over sequences of words. It allows predictions of the most likely next word given a number of previous ones, as well as language generation.

The simplest language model (LM) is an n-gram model, which assigns probabilities to sequences of n words (Katz, 1987; Brown et al., 1990; Bahl et al., 1983). This is equivalent to approximating the history of a word by the last $n-1$ tokens.

With the revival of distributed representations, a new model was needed to make the most of the subsymbolic power of dense words representations: neural LMs exploit the ability of neural networks to learn distributed representations (Paccanaro and Hinton, 2000) to reduce the problems of sparsity and high dimensionality (Bengio and Bengio, 2000). Bengio et al. (2003) showed that such distributed representations could be used as features for feedforward neural networks, with the result of outperforming n-gram models. The feedforward model, however, is still limited by its Markovian approximation of the history of a word.

Recurrent neural networks overcome this limitation (Mikolov et al., 2010) as their structure exploits the sequential nature of natural language. To ensure that the entire word history is truly considered, most neural LMs used today are gated recurrent neural networks such as the Long Short-Term Memory (Hochreiter and Schmidhuber, 1997).

As a result of the gradient flow properties of RNNs, recurrent neural LMs are known for their ability to capture long-term dependencies in natural language. However, it is still an open question to what extent recurrent models can capture long-term syntactic relationships. A model that aims at mimicking human-level language understanding needs to encode both short-term and more intricate, long-term syntactic phenomena. Linzen et al. (2016) made an attempt to scrutinise the question whether LSTM LMs capture a specific type of syntactic dependency, namely subject-verb agreement. The following sentences are examples of this phenomenon:

(1) The **student is** in the lecture hall.

(2) The **student** with his many books **is** in the lecture hall.

A LM should be able to assign a higher probability towards "is" in sentence (1) given its dependency on the singular subject "student". In this example, subject and verb are directly adjacent, but this need not be the case. In contrast, in sentence (2) there are intervening words between subject and verb that could erroneously influence the probability assigned to the verb with the correct number. A specific type of intervening words are agreement attractors. In the case of subject-verb

agreement, these are intervening nouns with the opposite number of the subject. In sentence (2), the noun "books" is the agreement attractor and, due to its appearance in plural form, it could cause the model to wrongly predict the number of the verb "to be" as plural instead of singular.

Linzen et al. (2016) showed that an increasing number of intervening words and agreement attractors leads to a higher error rate when the LSTM assigns probabilities to verbs in their singular and plural form. More specifically, Linzen et al. (2016) demonstrated that an LSTM LM without grammatical supervision performs worse than chance on a sample of sentences with agreement attractors. Their extensive error analysis is an attempt to obtain clear indicators on why and to what extent the LM failed to perform a task that requires grammatical information.

In this paper, we aim to go one step further in explaining the failure of recurrent neural LMs to adequately learn syntactic dependencies. The approach we use to analyse how grammatical structure is encoded in LMs is inspired by Section 5 in (Linzen et al., 2016), which reports the investigation on the capacity of an LSTM without grammatically relevant supervision to correctly assign probabilities in the case of subject-verb agreement. In addition to subject-verb agreement judgements, we experiment with verb form government as it is another simple variable-distance phenomenon that can serve as a further baseline requiring morphosyntactic information.

To present more quantitative evidence and possibly insights on the relationship between syntactically interdependent constituents, we employ a recently introduced gated RNN, the Recurrent Additive Network (RAN) (Lee et al., 2017). The RAN model was shown to achieve state-of-the-art performance in language modelling tasks while simplifying the architecture of the LSTM. The proposed simplification has the benefit that, in a word prediction task, it allows to identify and visualise to which extent previous words contribute to the current prediction.

The strongest motivation for this paper is the acknowledgement of the need and usefulness of explanatory work in the Natural Language Processing field, not in opposition to exploratory or competitive research but as a means of providing solid basis for exploration and performance tuning. Recurrent neural models are widely used nowadays with the assumption that they are able to capture long-term dependencies yet Linzen et al. (2016) demonstrated that the way subject-verb agreement is modelled by state-of-the-art LSTMs can be grammatically incorrect or at least counter-intuitive. Our aim is to effectively visualise how backward context influences the choice of the next most likely word in a sequence in order to explain the pitfalls of sequence-based models. Moreover, this work is inspired by the idea that understanding why grammar is incorrectly encoded in recurrent neural LMs is a necessary step towards the design of novel architectures that better fit both the sequential nature and the hierarchical structure of language.

In the remainder of this paper, we first introduce the two types of variable-distance dependency constructions which are relevant to this work (Section 2); then we describe the Recurrent Additive Network, the details behind RAN weight visualisation, and its limitations (Section 3); Section 4 presents how experiments were organised and the results are illustrated in Section 5; finally, we discuss our findings and propose ideas for future work in Section 6.

2 Variable-distance dependency constructions

Although the architecture of recurrent neural LMs allows capturing long-distance dependencies, it is more principled to use rather local phenomena as a starting point for diagnostic research on such models. We use the expression *variable-distance* dependencies to signal that it is not entirely clear where to draw a line between short and long distance dependency constructions. Due to linguistic creativity, words which are typically in a local relation can sometimes appear very distant from each other:

(3) **John**, who finally came back from Cuba after three long years, **is** visiting us tonight.

Ten words and two commas occur between "John", the subject, and "is", the syntactic head of the verb phrase.

In this section, we present two simple types of variable-distance dependencies: subject-verb agreement and verb form government. We will explain why these are relevant phenomena for

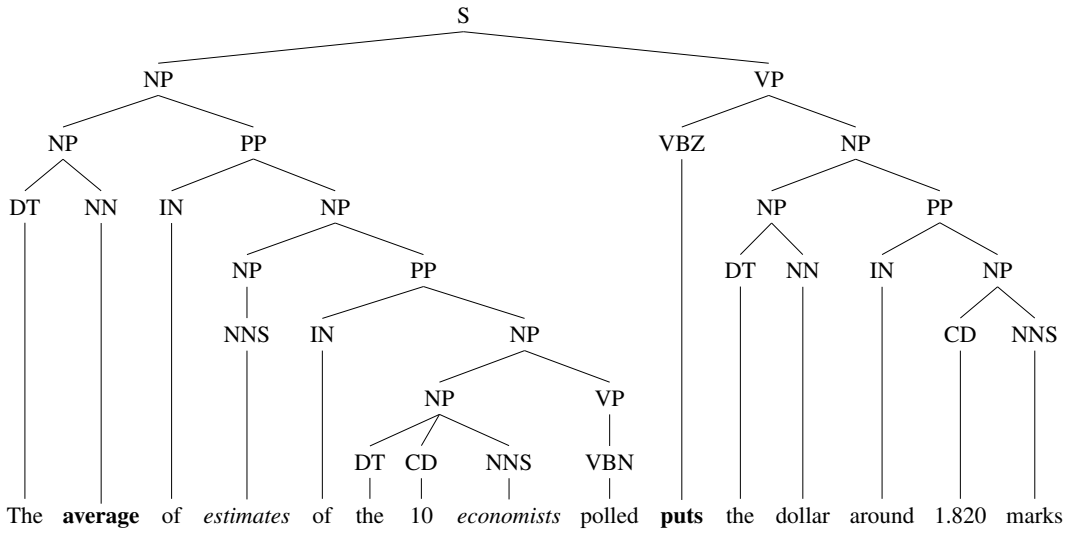


Figure 1: A constituency tree for one of the parses of (the main part of) sentence (10) obtained via the Stanford online parser (Klein and Manning, 2003).

analysing the properties LMs, and describe what difficulties they entail.

2.1 Subject-verb agreement

The agreement phenomenon occurs in the English language whenever a sentence contains a verb and a noun acting as its subject.

(4) The Paris **train leaves** at midnight.

(5) Steven's **children love** playing badminton.

In order for an English sentence to be grammatical, the verb and its subject need to *agree* in two properties: person (first, second, or third) and number (singular or plural).

(6) *Which **trains leaves** after midnight?

Sentences (4) to (6) are cases of local relations as the head of the subject is adjacent to the verb. Such short-distance dependencies seem easy to capture even via handcrafted rules. However, examples (3), (7), and (8) show that a variable number of intervening words can occur.

(7) **John**, who finally came back from Cuba, **is** visiting us tonight.

(8) **John**, after three long years, **is** visiting us tonight.

Without any syntactic analysis, it is not trivial for a neural LMs to identify all subject-verb pairs in a sentence as they might appear at any distance from each other. Moreover, intervening words can themselves be candidates for agreement. For example nouns can appear between the subject and the verb and lead the model to agreement attraction errors (Bock and Miller, 1991). Following the terminology used in Linzen et al. (2016), we refer to nouns with the opposite number to the subject as *agreement attractors* (italicised words):

(9) No **price** for the new *shares* **has** been set.

Some cases of multiple agreement attractors might even confuse English speakers:

(10) The **average** of *estimates* of the 10 *economists* polled **puts** the dollar around 1.820 marks at the end of November and at 141.33 yen.

This sentence has two agreement attractors that set subject and verb 7 tokens apart. Syntactic knowledge is necessary in order to identify "average" as the head of the noun phrase acting as a subject for the verb "puts" (see Figure 1). That is, a LM that wants to capture subject-verb agreement needs to be able to recursively collapse the sentential material between subject and verb into the noun phrase headed by the subject. Only in a syntactically-informed representation (such as a parse tree) will the agreeing words result adjacent to each other. Similarly, in the context of dependency grammars, the model needs to be able to detect that there is a direct dependency relation between "puts" and "average" (see Figure 2).

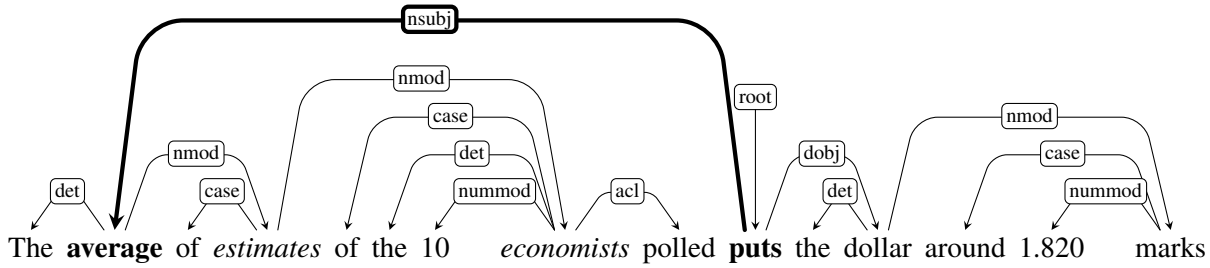


Figure 2: A dependency tree for one of the parses of (the main part of) sentence (10) obtained via the Stanford online parser (Chen and Manning, 2014).

Unlike parsers, LMs do not consider language in its hierarchical structure and thus they lack the traditional means of recognising syntactically related words. Recurrent LMs can nevertheless effectively encode sequential information, and in this paper we attempt to observe the extent to which the sequential model can approximate a (possibly recursive) hierarchical structure.

2.2 Verb form government

Government is another variable-distance phenomenon. It is similar to agreement but postulates that one word, the *governor* selects some grammatical feature of another word. A typical example of government in English is the control by verbs and preposition of the grammatical case of their complement:

(11) a. She sees him.
b. *She sees he.
c. *Her sees him.
d. ?Her sees he.

Grammatical case is encoded in English only in pronouns. As verbs tend to realise personal pronoun arguments locally (example (11)) and, in contrast, interrogative and relative pronouns sometimes move very far from their syntactically natural position (*wh*-movement), we select a simpler type of government for our experiments: verb form government.

(12) a. The sun **is rising**.
b. *The sun **is rise**.

(13) a. The time **has come**.
b. *The time **has comes**.

In complex verb phrases, i.e. phrases composed by multiple verbs, the auxiliary verb selects the grammatical properties of the main verb. In particular, the former selects a *verb form* of the latter. Sentence (12-a) shows that the selected verb form for "to rise" is the present participle; in sentence (13-a), the auxiliary word "has" selects a complement in past participle form. Additional verb forms are the base form and third-person singular present form (for regular verbs).

As in the case of subject-verb agreement, neural LMs may not be able to straightforwardly identify government phenomena in a sentence without explicit syntactic analysis as the governor and the governee can be separated by sentential material. Again, intervening words can themselves be candidates for the role of governor. For example, verbs can appear between the auxiliary and the main verb and lead the model to attraction errors. We refer to verbs with a different verb form than that of the governor as *government attractors* or simply *attractors* (again, italicised):

(14) Chief executives and presidents **had** *come* and **gone**.

Here, the attractor is a governee. In more involved examples, also nouns can act as attractors if they happen to be ambiguous:

(15) The French company **has** *expanded* its bottled water selection in some *stores* and **added** fresh juice in some outlets.

Before concluding this section, it is worthwhile to mention a difference between the subject-verb agreement and verb form government phenomena. On the one hand, subjects and agreement attractors are mostly nouns—hence content words. In contrast, syntactic heads of verb phrases, i.e. governors, are often auxiliary verbs—which can be considered as function words—and government attractors are often main verbs, thus content words.

Finally, we acknowledge that the distinction be-

tween government and agreement is not always straightforward, yet a comparison of these two linguistic categories is outside the scope of this paper.

3 Methodology

To examine the behaviour of neural LMs on the linguistic phenomena described in Section 2, we use a Recurrent Additive Network (RAN) (Lee et al., 2017). A RAN cell is essentially the classical LSTM cell with the internal nonlinearity and the output gate removed:

$$\begin{aligned}\tilde{\boldsymbol{c}}_t &= \boldsymbol{W}_{cx}\boldsymbol{x}_t \\ \boldsymbol{i}_t &= \sigma(\boldsymbol{W}_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ix}\boldsymbol{x}_t + \boldsymbol{b}_i \\ \boldsymbol{f}_t &= \sigma(\boldsymbol{W}_{fh}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{fx}\boldsymbol{x}_t + \boldsymbol{b}_f) \\ \boldsymbol{c}_t &= \boldsymbol{i}_t \circ \tilde{\boldsymbol{c}}_t + \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} \\ \boldsymbol{h}_t &= g(\boldsymbol{c}_t),\end{aligned}$$

where $\boldsymbol{x}_t$ is the input vector at time t, $\boldsymbol{i}_t$ and $\boldsymbol{f}_t$ are the input and forget gates, $\boldsymbol{c}_t$ is the state of the RAN cell, and $\boldsymbol{h}_t$ is the output vector.
The main motivation for using RAN is that the state of a RAN cell $\boldsymbol{c}_t$ can be expressed as a sum of linearly transformed inputs $\{\boldsymbol{x}_0, ..., \boldsymbol{x}_t\}$:

$$\begin{aligned}\boldsymbol{c}_t &= \boldsymbol{i}_t \circ \tilde{\boldsymbol{c}}_t + \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} \\ &= \sum_{j=1}^{t} \left(\boldsymbol{i}_j \circ \prod_{k=j+1}^{t} \boldsymbol{f}_k\right) \circ \tilde{\boldsymbol{c}}_j \\ &= \sum_{j=1}^{t} \boldsymbol{w}_j^t \circ \tilde{\boldsymbol{c}}_j.\end{aligned}$$

For the language modelling task, this implies that each RAN state $\boldsymbol{c}_t$ is a weighted sum of word embeddings of all the words from the start of the sequence up to the t-th word. This allows us to explicitly trace how much each of the RAN states was influenced by all inputs preceding it, and for example see which previous word has the strongest influence in predicting the next word at timestep t:

$$v_t = \arg\max_{j=1}^{t-1} \left(f(\boldsymbol{w}_j^t)\right), \tag{1}$$

where $f(\cdot)$ is a function that computes the influence of a particular $\boldsymbol{w}_j^t$. A theoretically sound choice of $f(\cdot)$ from i.e. $\max(\cdot)$, L1 or L2 norms of $\boldsymbol{w}_j^t$ is an area for future investigation.

Unfortunately, this property of RAN is of somewhat limited use for analysing various linguistic phenomena. Essentially, it allows to observe the relative magnitude of how much the model remembers each of the previous words. However, it does not allow to conduct an experiment of the following form: given a sequence of words $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_{t-1}\}$ and several potential candidate words $\{\boldsymbol{x}_t^0, ..., \boldsymbol{x}_t^n\}$, see which word from $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_{t-1}\}$ has the largest influence on predicting a particular $\boldsymbol{x}_t^i$. For example, it is not possible to get a result like: in the sentence "There is a **job** that *they* think **is / are** ...", the model predicts **are** when the word *they* is more influential than the word **job**, and predicts **is** otherwise. This is the case since $\arg\max_{j=1}^{t-1} \left(f(\boldsymbol{w}_j^t)\right)$ will be the same for each $\boldsymbol{x}_t^i$.

4 Experiments

4.1 Model training and data collection

For our experiments we used a Pytorch (Paszke et al., 2017) implementation of RAN by Heinzerling (2017). The model was trained on the Penn Treebank Corpus (Marcus et al., 1993) for 200 epochs (see Appendix for more detailed information about training).

In order to obtain sentences containing subject-verb agreement and verb form government phenomena, the PTB corpus was queried using regular expressions and specific POS-tags patterns. The selection of sentences was then organised into three categories. For the case of subject-verb agreement, the *adjacent* category contains sentences where the verb follows the subject directly. The *intervening* category includes sentences where at least one word occurs between subject and verb, but no agreement attractors interfere. The *attractor* category consists of sentences that have an agreement attractor. In the case of verb form government, the first category contains sentences where the governor is directly adjacent to the verb, in the second category non-attractor words intervene, and in the third at least one agreement attractor is present. These examples are the result of manual inspection. Due to time constraints, we only considered a relatively small number of sentences (see Table 1).

4.2 Grammaticality judgements

The first step of the analysis was an attempt to replicate the findings presented in Linzen et al. (2016) that for sentences with subject-verb agreement phenomena the LM would predict the wrong verb number more often in sentences containing

	adjacent	intervening	attractor
Subj-verb	17	12	6
Verb form	19	15	14

Table 1: Number of dependency construction examples per category.

attractors than in those from the *adjacent* or *intervening* categories. Hence, we compared perplexities of grammatical sentences (with the correct verb person and number) and ungrammatical sentences (with cases of failed agreement), and counted how many times the trained RAN model would predict the correct verb form.

Additionally, to extend our analysis to other variable-distance construction (as Linzen et al. (2016) encouraged), we performed the same experiment with the dataset of verb form government sentences. In this case, ungrammatical sentences are those where the *governed* verb appears in a different form than that selected by its governor.

4.3 Error analysis

The previous section described how we quantify in how many sentences the RAN model assigns a higher probability to ungrammatical sentences. In the case of subject-verb agreement, the model makes an error if it assigns a higher probability to sentences where the verb does not agree in person and number with the subject. In verb form government, on the other hand, the model assigns probabilities to n sentences, where n is the number of alternative verb forms and it varies according to the regularity or irregularity of the lemma.

To aid the sentence-by-sentence manual analysis we used a visualisation of the L1 norm of the weights w_j^t corresponding to previous words in the sentences as outlined in the methodology section. With this visualisation technique, the influence e.g. of the subject or the *governor* at the time step of the verb prediction was scrutinised and compared to other words in the sentence. We also investigate whether visualising history-word activation in general can help in such an error analysis.

4.4 RAN weights and syntactic information

With the introduction of weight visualisations for the RAN model (Anonymous, 2018), an interesting trend was observed, namely that the L1 norm of the weights for content words was initially stronger and decreased slower over time steps compared to the weights of function words. That is, content words seem to be "remembered" longer by the RAN model and therefore their signal can exert stronger influence in predicting the next word over more steps.

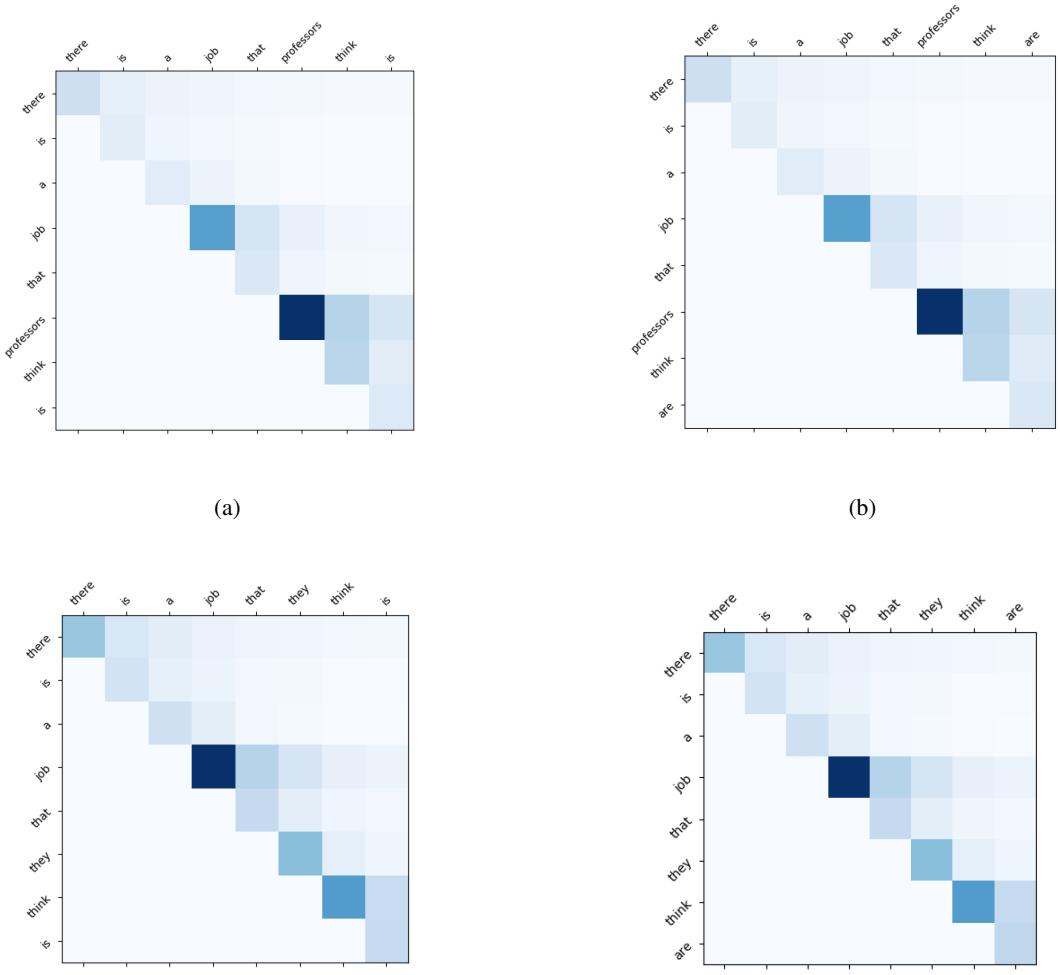
As the RAN model appears to remember content words over longer distances, we attempted to detect relevant differences between content and function words acting as an agreement attractor. If the attractor is a function word, we would expect that it does not influence the prediction of the number of the verb very actively. In contrast, we would expect that in sentences in which an agreement attractor is a content word the model fails more often at recognising the sentence grammaticality. In the case of subject-verb agreement, the attractor can be manually transformed from noun into pronoun—hence from content to function word. Consider the following sentences:

(16) a. There is a **job** that *professors* think **is** interesting.
b. *There is a **job** that *professors* think **are** interesting.

(17) a. There is a **job** that *they* think **is** interesting.
b. *There is a **job** that *they* think **are** interesting.

The sentences are equal with the exception of the agreement attractors "professors" and "they". In such examples, we consider the model to be wrong if it assigns lower perplexities to the sentences when the verb is "are" (sentences (16-b) and (17-b)) instead of "is" (sentences (16-a) and (17-a)). Linguistically, it should make no difference whether the agreement attractor is a noun or a pronoun. It is, however, an open question whether the predictions of the model are more often incorrect when the attractor is a content word.

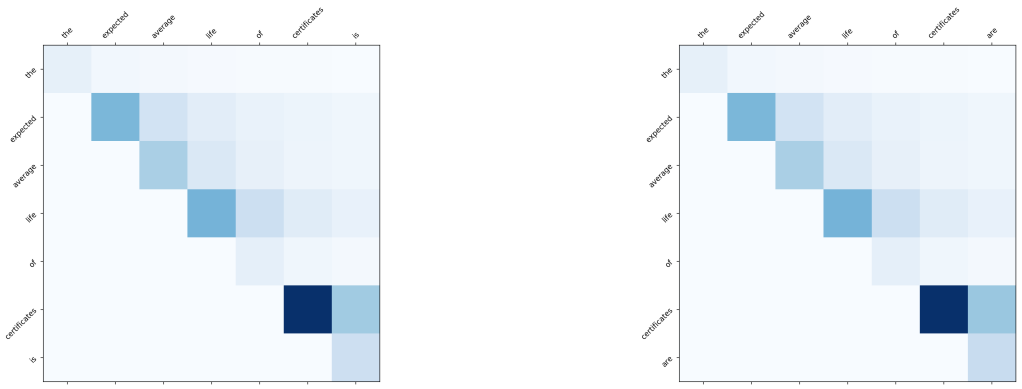
Furthermore, there remains the possibility that even if function words act as attractors and their influence on the the crucial time step of the model is very low, the model still reproduces the same error. Again, due to time constraints, we have manually inspected a few examples to test this eventuality.



(a) (b)

(c) (d)

Figure 3: Weight visualisations using Equation 1 and L1 norm.



(a) PP = 5303 (b) PP = 6547

Figure 4: Weight visualisations using Equation 1 and L1 norm. Original sentence: The expected average life of the certificates is N years with the final scheduled payment in October N.

	adjacent	intervening	attractor	random
Subj-verb	76.5.9	33.3	33.3	50.0
Verb form	57.9	53.3	35.7	23.2

Table 2: Accuracy of RAN grammaticality judgements.

5 Results

5.1 Grammaticality judgements

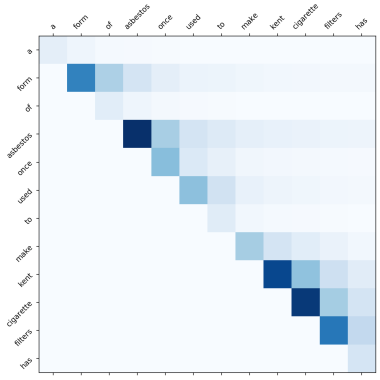
Comparing the perplexities of grammatical sentences and ungrammatical sentences (i.e. instances of failed agreement or government), we computed how often the RAN model made the correct grammaticality judgement. Table 2 presents the model accuracy for each of the six sentence categories (phenomenon: agreement or government; distance: *adjacent*, *intervening*, *attractor*). In line with Linzen et al. (2016), we find that for subject-verb agreement the model achieves the lowest grammaticality judgement performance in the case of *attractor* examples compared to *adjacent* and *intervening* examples. Additionally, we provide further evidence for the generality of this performance in that we replicate it with the phenomenon of verb form government.

5.2 Error analysis

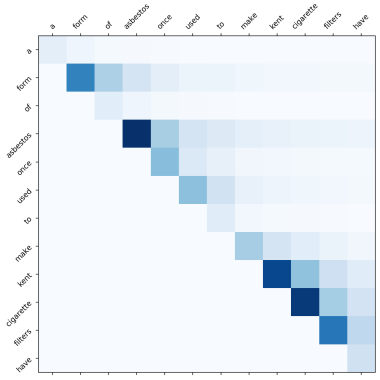
In this subsection, we focus on sentence pairs for which the RAN model makes an incorrect grammaticality judgement. In general, we observe that the further apart two syntactically related words are, the higher the error rate. Also it seems that errors are more likely to happen the closer the attractor is to the attractee (see e.g. Figure 5). However, there are also examples where this is not the case, as in Figure 4. In this sentence, the attractor "certificates" is directly adjacent to the verb—and the plot shows that "certificates" is very influential—but the model still correctly assigns a lower perplexity to the grammatical sentence.

Furthermore, we observe that content words are in general more active and function words less active in terms of L1 norm of the weights. However, although subjects and agreement attractors are mostly nouns—therefore mostly content words—these rather high activations, which are remembered for many timesteps, do not seem to particularly aid the model in assigning higher probability to correctly inflected verbs.

In the context of verb form selection, governors are auxiliary verbs, which are function words.



(a) PP = 5771



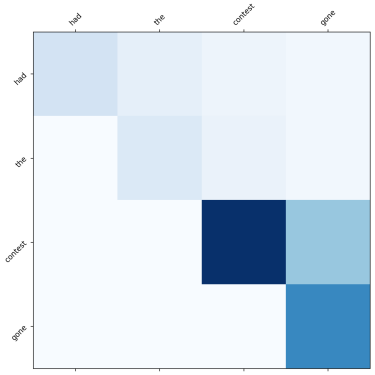
(b) PP = 5446

Figure 5: Weight visualisations using Equation 1 and L1 norm. Original sentence: A form of asbestos once used to make Kent cigarette filters has caused a high percentage of cancer deaths among a group of workers exposed to it more than N years ago researchers reported.

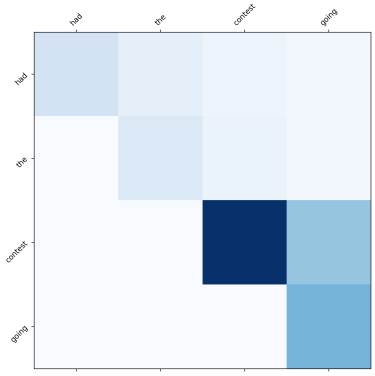
Therefore they are remembered by the model for a shorter time and have less low weight activations. From our manual inspection it appears, however, that the weight activation of the governor does not influence the choice of the verb form. Consider for example Figure 6, where the model correctly assigns a lower perplexity to the sentence fragment containing "gone". In a seemingly simpler case (see Figure 7), the model assigns a lower perplexity to the sentence containing "has voting" rather than "has voted". To conclude, there appears to be no clear link between L1 norm of weights and syntactically aware decisions.

5.3 RAN weights and syntactic information

The weight plots and corresponding perplexities of six partially handcrafted sentences were analysed manually. Interestingly, when comparing



(a) PP = 4886



(b) PP = 5152

Figure 6: Weight visualisations using Equation 1 and L1 norm. Original sentence: Had the contest gone a full seven games ABC could have earned an extra N million dollars in ad sales on the seventh game alone compared with the ad take it would have received for regular prime-time shows.

perplexities of sentence (16-a) versus sentence (16-b) and sentence (17-a) versus (17-b) respectively, the RAN model correctly assigns lower perplexities to the sentence with the correct verb "is" (Perplexities: PP(16-a) = 7730, PP(16-b) = 8155, PP(17-a) = 6257, PP(17-b) = 6497). The plots for this example can be found in Figure 3. A few interesting observation can be made, for example that the content word "professor" indeed has a higher initial L1 norm of its weights compared to the L1 norm of the weights for the function word "they" in the same position. Also, the L1 norm of the weights of the subject "job" is higher in case the attractor is "they". Crucially, in case the attractor is "they" the L1 norm of the weights of the subject "job" is higher compared to the attractor L1 of the weights at the last time point (i.e. where the verb is predicted). The observation that if the attractor is a function word the subjects L1 norm of weights is slightly higher in the last time step was observed for our other test sentences as well. However, the RAN model always assigned the lower perplexity to the same outcome (either correct or incorrect).

6 Conclusions and future work

In this paper, we analysed the ability of a RAN language model to capture variable-distance syntactic dependencies. Inspired by the analysis in (Linzen et al., 2016), we tested the RAN on grammaticality judgements on instances of subject-verb agreement and verb form government. To enhance our analysis, we visualised the RAN weights using different ways of quantifying history-words activations.

In accordance to previous studies, we observed that content words retain a high weight activation for a larger number of timesteps than function words. Indeed the model seems to selectively activate salient (often infrequent) words that define the semantic context of the sentence. However, we concluded that the magnitude of RAN weight activations does not seem to provide the model with clear indications concerning how to assign probabilities to a grammatical form over an ungrammatical one. In other words, high activation for subjects or auxiliary verbs does not guarantee correct grammaticality judgement, and high attractors activation does not guarantee incorrect grammaticality judgement.

Given the limitations of our analysis, we cannot exclude that the RAN model is syntactically aware. We nevertheless report a low accuracy on grammaticality judgements, as well as the surprising occurrence of elementary prediction errors (on very short sentences with no attractors). Hence we come to the conclusion that although recurrent LMs are widely assumed to *be able to* capture long-term dependencies, they cannot confidently analyse *syntactic* dependency constructions and therefore they often make syntactically unaware predictions. The sequential approximation that the recurrent LM uses to encode hierarchical structure is not linguistically satisfactory.

In future, we plan to collect a dataset of sentences containing different types of dependency constructions such as case government (see Section 2. Such a dataset would allow standard-

ised evaluation of LMs with respect to grammaticality judgements, and more effective diagnostic studies on the properties of LMs. Equipped with such a resource, we would like to replicate the experiments presented in this paper with a larger amount of annotated data. A further valuable contribution might emerge from quantifying how fast history words activations decrease: in particular, it is interesting to detect whether content words are remembered significantly longer than function words. Finally, it is worth experimenting with attention mechanisms (Bahdanau et al., 2014) in order to see if they can help recurrent models to identify syntactically relevant (Vinyals et al., 2014) history words.

Team responsibilities

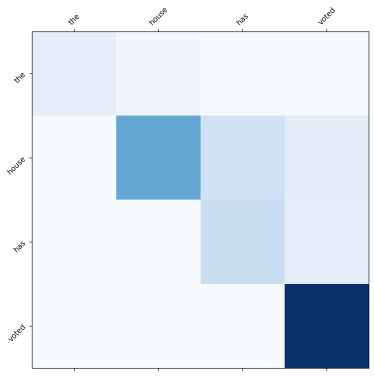
- *Florian*: implementation of feed forward NN (Bengio), linguistic annotation, error analysis
- *Dmitrii*: math insights, training the model, finding and fixing bugs, visualisation
- *Mario*: project idea, linguistic annotation, error analysis, corpus querying
- *Everyone*: coding up the analysis, writing the report

Appendix

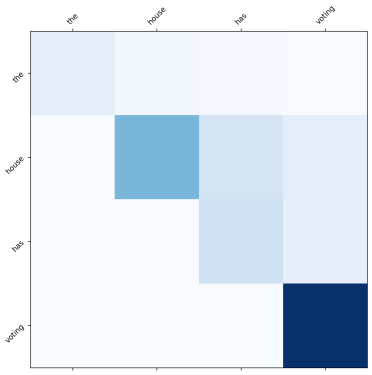
RAN training. These are the main training parameters for the RAN language model:

- batch size: 64
- epochs: 200
- dropout: 0.2
- learning rate: 0.01
- maximum sequence length: 35

Code. The code for this paper is open-source and available at `github.com/Procope/ran-lm-eval`.



(a) PP = 4474



(b) PP = 3742

Figure 7: Weight visualisations using Equation 1 and L1 norm. Original sentence: The house has voted to raise the ceiling to N trillion dollars but the senate isn't expected to act until next week at the earliest.

References

Anonymous. 2018. Long short-term memory as a dynamically computed element-wise weighted sum. *International Conference on Learning Representations* `https://openreview.net/forum?id=HJOQ7MgAW`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. `http://arxiv.org/abs/1409.0473`.

Lalit R Bahl, Frederick Jelinek, and Robert L Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE transactions on pattern analysis and machine intelligence* (2):179–190.

Samy Bengio and Yoshua Bengio. 2000. Taking on the curse of dimensionality in joint distributions using neural networks. *IEEE Transactions on Neural Networks* 11(3):550–557.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Kathryn Bock and Carol A Miller. 1991. Broken agreement. *Cognitive psychology* 23(1):45–93.

Peter F Brown, John Cocke, Stephen A Della Pietra, Vincent J Della Pietra, Fredrick Jelinek, John D Lafferty, Robert L Mercer, and Paul S Roossin. 1990. A statistical approach to machine translation. *Computational linguistics* 16(2):79–85.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 740–750.

Benjamin Heinzerling. 2017. A pytorch implementation of recurrent additive networks. https://github.com/bheinzerling/ran.

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

Slava Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE transactions on acoustics, speech, and signal processing* 35(3):400–401.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 423–430.

Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2017. Recurrent additive networks. *CoRR* abs/1705.07393. http://arxiv.org/abs/1705.07393.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *arXiv preprint arXiv:1611.01368* .

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics* 19(2):313–330.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*. volume 2, page 3.

Alberto Paccanaro and Geoffrey E Hinton. 2000. Extracting distributed representations of concepts and relations from positive and negative propositions. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*. IEEE, volume 2, pages 259–264.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch .

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. 2014. Grammar as a foreign language. *CoRR* abs/1412.7449. http://arxiv.org/abs/1412.7449.