

# Empirical Evaluation of LSTM Language Models on Syntactic Dependencies

**Maartje de Jonge**

Student ID: \*\*\*  
email@domain

**David Rau**

Student ID: 17725184  
david.rau@student.uva.nl

## Abstract

- main idea - key findings

## 1 Introduction

Neural networks have become increasingly popular for the task of language modeling. While feed-forward networks only take into account a fixed history of preceeding words to predict the next word, standard recurrent neural networks (RNN) and Long Short-Term Memory (LSTM) network architectures have the ability to capture long-distance statistical regularities. The access to potentially unlimited history has resulted in substantial improvements in perplexity and error rates compared to feed-forward networks (Mikolov *et al.*, 2010; Sundermeyer *et al.*, 2013).

Regularities in natural language are often sensitive to syntactic structure. RNNs and LSTMs are sequence models that do not explicitly incorporate syntactic structure, capturing such dependencies is therefore challenging. (Linzen *et al.*, 2016) investigate the capability of LSTMs to learn syntactic dependencies, taking number agreement in noun-verb dependencies as an example. Their results show that the language modeling objective by itself is not sufficient for learning structure-sensitive dependencies; and that more explicit supervision is required.

In this paper we investigate an LSTM language model in more detail to get a better insight into what information these models actually encode. We consider statistic, syntactic and semantic information and analyse how the model uses this information to establish number agreement. We take an empirical approach. That is, we treat the model as a black box and learn about it by observing its behavior in carefully designed experiments.

In line with the results of (Linzen *et al.*, 2016), our model was able to establish number agreement

for most of the simple sentences, i.e. sentences without intervening nouns. This shows that our model is able to learn the plurality number for nouns and verbs. A detailed analysis of the errors made in simple cases showed that they could be explained partially by statistics on the training data, i.e. the frequencies of nouns and the frequency ratio of plural versus singular verbs.

As expected from (Linzen *et al.*, 2016), our model performed worse-than-change on complex sentences with intervening nouns of opposite number. The additional challenge in this case is to identify the head of the subject without being distracted by other structurally irrelevant nouns. The head of the subject is typically implied by the syntactic structure of a sentence; e.g. the sentence “The toys of the boy lay” requires a plural verb, while the sentence “The toys that the boy plays with” calls for a singular verb. We experimented with various syntactic templates to see if the model shows sensitivity to syntactic structure. The results suggest a modest sensitivity to syntactic information exposed by function words.<sup>1</sup>

## 2 Related Work

Our work builds on the results described in (Linzen *et al.*, 2016). We summarize this paper below.

LSTMs are sequence models that can capture long distance statistical regularities, but do not have built-in hierarchical representations. Linzen *et al.* investigate whether LSTMs are able to capture dependencies that follow from syntactic structure. More specific, the paper investigates number

---

<sup>1</sup> Semantic information can also help the model to identify the structurally relevant noun. For example, in the sentence “The prices of the product ... [stabilize/stabilizes]”, prices are more likely to stabilize than products. Unfortunately, the planned experiment on semantic information was not carried out because one of the team members quite the project.

agreement in English subject-verb dependencies as an example of a structure sensitive dependency.

The paper compares the performance of LSTMs trained with an explicit grammatical target as training objective, as well as a more generic language model trained with the target to predict the next word. The grammatical models had the following objectives: *number prediction* is trained to predict the number of the next verb without knowing the verb, *verb inflection* is similar to number prediction but with the difference that the model knows the verb, for *grammatical judgement* the network is asked to judge whether a sentence is grammatical correct, without knowing the position of a possible incorrect verb. The models are trained on a corpus without syntactic annotations (Wikipedia). The models are evaluated on real sentences taken from this corpus that were sampled based on their grammatical complexity.

All models achieved an overall error rate below 7%. However, the explicitly trained models perform much better (verb inflection 0.8%, number prediction 0.83%, grammatical judgement 2.5%) compared to the language model (6.8%). The overall high accuracy for all models can be explained by the fact that most naturally occurring sentences are actually simple, that is, no intervening nouns between verb and subject.

The differences between the models become more pronounced when evaluating them on grammatically complex sentences. The performance of the grammatically trained models degrades slowly, achieving error rates below 20% even with four intervening nouns of opposite number (attractors). The language model at the other hand performs worse-than-chance on most complex cases. The worse-than-chance performance indicates that the intervening nouns actively confuse the language model. Repeating the experiment with a state of the art language model (Józefowicz *et al.*, 2016) does not change the picture, the state-of-the-art language model performs poorly on complex sentences compared to the explicitly supervised models.

The accuracy of the grammatically trained models strongly suggests that the model is sensitive to structural dependencies. To find additional support for syntactic sensitivity the authors inspected the inner workings of the number prediction model. Principal component analysis on embeddings of singular and plural nouns shows

that the first principal component corresponds almost perfectly to the expected number of the noun, which suggests that the model learns these numbers very well. Analysis of activations in response to specific grammatical constructs reveals units that store relevant structural information such as the number of the main clause subject, the number of the most recent noun of the current phrase and the embedding status.

The authors conclude that LSTMs can capture grammatical structure given targeted supervision; while the language modeling objective of predicting the next word is insufficient. They advice to supplement language modeling objectives with more explicit targets for tasks in which it is desirable to capture syntactic dependencies.

### 3 Replication

#### 3.1 Singular and Plural Nouns

**Data:** The language model was tested on lower case sentences that were generated from the Wall Street Journal section of the Penn Treebank (Marcus *et al.*, 1993). Therefore, 40 nouns and verbs, that were amongst the most common and occur in the test corpus as well as in language model's corpus, were extracted. Those words build the base for the sentence generation for the subsequent experiments.

**Model evaluation:** In order to evaluate the performance of our model, we query it with the sentence containing both, the plural (1) and the singular (2) mode of the verb:

the producer plan (1)

the producer **plans** (2)

According to the experiments in (Linzen *et al.*, 2016) we examine the models error rate predicting the number of a verb in a sentence when nouns intervene between the subject and the verb.

We next test how the model's ability to predict the number of the verb is affected by none and one intervening noun, respectively. If there is an intervening noun we keep track whether the number of the noun differs from the number of the subject. If it does so it is referred to as an *agreement attractor*. In this way we can easily spot whether the model makes use of the most obvious heuristic: choosing the the number of the verb only in dependence of the last intervening noun in the sentence.

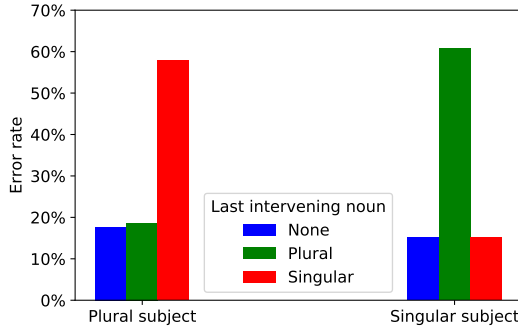


Figure 1: Error rates of the language model plotted against count of attractors in dependencies with homogeneous intervention.

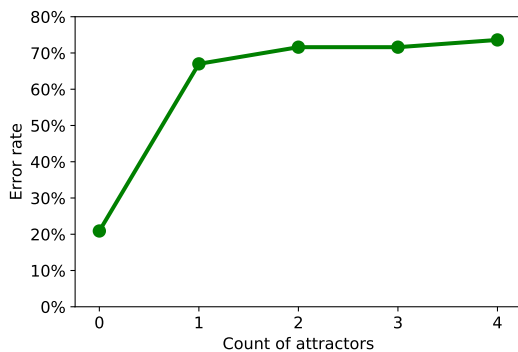


Figure 2: Error rates of the language model plotted against: presence and number of last intervening noun

As depicted in Figure 1 our model performs slightly worse for plural subjects (17.7% error rate) than for singular (15.2% error rate) when no intervening nouns are present. An intervening noun with the same number as the subject causes a slight increase of the error rate to 18.6% and a decrease to 15.1%, respectively. Surprisingly, when the number of the subject differs from the intervening noun the error rates increased dramatically; in singular subjects to 57.9% in plural subjects to 60.8%. The fact that it performs worse than predicting the number by chance implies that the model indeed predicts the number of the verb in dependence of the last noun and therefore fails to find the dependency between verb and noun.

### 3.2 Multiple Attractors

In the following we examine the error rate when adding multiple attractors to the sentence. In order to avoid the model of being distracted by an intervening noun with same number as the subject we only insert nouns with the same number. (Linzen *et al.*, 2016) refer to such as *dependencies with homogeneous intervention*. In (3) the underlined represent homogeneous interventions, whereas in (4) the number of the intervening nouns differ. The bold words highlight the dependency between noun and corresponding verb.

the **interest** in the shares of the businesses **rises** ...  
(3)

the **interest** in the shares of the *business* **rises** ...  
(4)

Figure 2 shows that with an increasing number of noun interventions the error rate goes from 20.8% (0 attractors) up to 73.6% (4 attractors) which is close to randomly guessing the number of the verb.

## 4 Own Experiments

To predict the correct number of a given verb, the language model should be able to 1. identify the noun that is the head of the subject for the verb 2. establish the number of the noun (non-trivial since no knowledge of -s postfix for plurals) 3. establish the number of the given verb forms (also non-trivial).

In the first sub section we investigate if our model is able to do this for simple cases with only a single noun in the prefix. In the second sub section we investigate if our model can handle more complex cases with two nouns in the prefix, and

what information it then uses to identify the head of the subject.

#### 4.1 Noun-Verb Agreement in Simple Cases

In this section we further investigate the ability of the model to establish number agreement for nouns and verbs in the simplest case following the pattern: “The  $\text{;noun}_i$   $\text{;verb}_i$ ”. Notice that the determiner “The” clearly indicates the position of the subject. From this we hope to get insights whether or not the model is sensitive to noun-verb agreement, i.e. if the model tends to predict plural verbs for plural nouns and singular verbs for singular nouns.

Using the 40 most frequent nouns and verbs that we already used earlier in the replication section, we generated 80 x 40 sentences. To evaluate how the language model is going to perform on these sentences we tested, for each noun-verb combination, how likely the model is going to predict the singular and the plural verb respectively.

As a result of this evaluation we obtained a matrix in which the entries indicate the models preference for the singular/plural verb. We calculate each value

$$v = P(< VBP >) / (P(< VBP >) + P(< VBZ >)), \quad (5)$$

where  $P$  is the models prediction probability,  $\text{;VBP}_i$  is the plural verb and  $\text{;VBZ}_i$  the singular verb. In this manner the matrix provides a ratio for singularity ( $v = 0$ ) and plurality ( $v = 1$ ). We call  $v$  the *plurality preference*. The upper part of the matrix contains all noun-verb combinations for the singular nouns, whereas the lower part contains the plural nouns.

In order to get a better visual impression we sorted the columns after the sums of the columns. The rows were sorted in a similar manner, but separately in order to conserve the separation between singular (upper half) and plural (lower half) nouns. The resulting matrix is shown in Fig. 4.1.

As we expected the majority of the verbs in the upper half of the matrix are predicted correctly to be singular with a high certainty. The same for the lower, plural half. Overall it can be said that around the horizontal axis as well as at the vertical borders we can observe that there are specific verb/noun combinations for which the model is rather uncertain. This also matches our expectations that there are regions in the matrix where the model is not clearly in favour of the singular

or plural verb.

In the following we will investigate the two special cases of verbs and nouns where the model is less certain: In the columns close to the left border of the matrix, we can find verbs of which the model prefers the plural verb over the singular, regardless the number of the subject. For example the verb ‘buy’ in column 0 has a plurality preference of 0.95. In the far right columns we find the verbs of which the plural verb is preferred over the singular. The verb ‘say’ in column 39 has a plurality preference value of 0.1625.

There are also nouns which clearly prefer one number over another. Around the middle of the matrix we can observe a few words that are clearly predicted wrong most of the times. The noun ‘income’ with a plurality rate of 0.15 is mostly predicted singular whereas for example the words ‘weeks’ (0.9) column 40, ‘months’ (0.9) column 41, ‘years’ (0.85) column 43, ‘days’ (0.7) column 44, are predicted plural most of the times.

To investigate the impact of the frequencies in the training data on the plurality rate (and therefore on the error) further, we calculated the error rate for each noun and plotted it against the frequency of the same noun. We expected a high error rate for low frequent nouns. As the Fig. 5 shows there is no clear evidence that a low frequency of singular or plural nouns in the training data leads to wrong predictions of the model. Figure 5 shows that there are singular as well as plural nouns with a low count that are predicted correctly most of the times (error rate  $\leq 0.2$ ).

The previous experiment neglects the fact that even though a word is frequent in the corpus it could never be the subject in the training sentence. We investigated whether the nouns that we encountered before as being predicted wrong most of the times, namely: ‘weeks’, ‘months’, ‘years’ and ‘days’ are in fact rarely the subject in the training data. This suggests that the model is indeed not able to establish the number agreement for nouns and verbs in a more general but rather in a statistical way. Figure 4 supports this assumption. It shows the plurality rate plotted against the plurality preference for each verb. We can see that with an increasing plurality rate the plurality preference increases, as expected. The line in the figure was generated by performing a linear regression through the points.

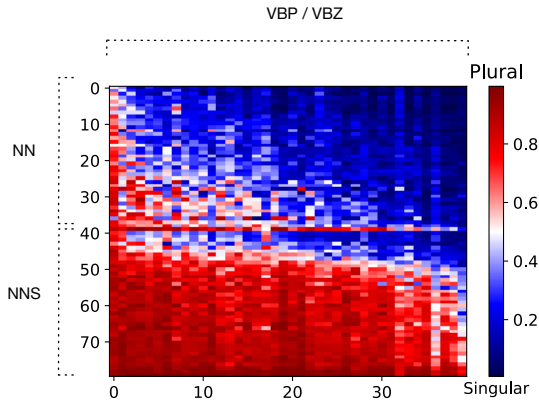


Figure 3: Singularity/plurality ratio for 80 x 40 sentences (generated from the most frequent 40 nouns and verbs). The upper half contains the singular nouns (NN), the lower half contains the plural nouns (NNS). A red cell indicates a plural preference for a specific word/noun combination, whereas a blue cell indicates a singular preference.

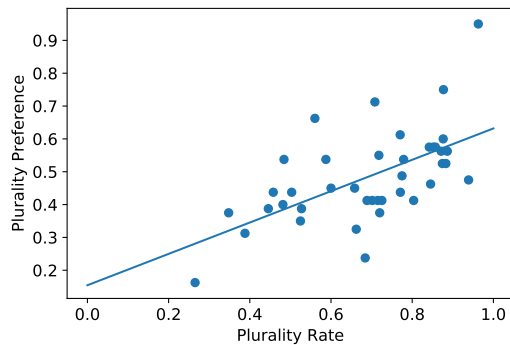


Figure 4: Plurality rate plotted against the plurality preference of the model for all nouns. The line arises from a linear regression through the points.

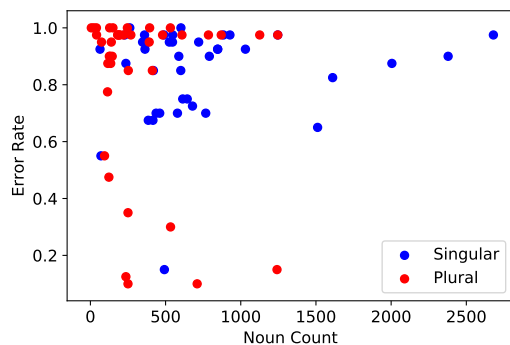


Figure 5: Influence of the noun frequency on the error rate of the model.

## 4.2 Noun-Verb Agreement in Complex Cases

In Section 3 we analysed the performance of the model on complex sentences, containing one or more nouns. The results show that the model is very sensitive to the most recent noun, performing worse-than-chance with only one single attractor.

In this section we investigate whether syntactic and semantic information can still help the model to establish number agreement in case of multiple nouns. We focus on sentences with exactly two nouns of opposite number.

### 4.2.1 Syntactic Information

Function words such as ‘that’ or ‘of’ carry important information about the syntactic structure of a sentence. In this Section we investigate if the model uses this information to establish number agreement for complex sentences.

We generated sets of sentence prefixes using different syntactic templates. An example is “The \_ of the \_.” We instantiate these templates by randomly picking two nouns and a verb from a set of frequently used nouns and verbs. Each combination of nouns and verbs instantiates two prefixes that differ by their plurality. For example “The company of the governments” and “The companies of the government”, with the option to choose between the verb forms ‘know’ and ‘knows’. Since the nouns and verbs are randomly picked, the generated prefixes are typically not semantically meaningful.

We generated 2 x 1000 sentences per template, for a total of 11 templates. The sentences for each template are constructed using the same noun and verb combinations. We defined 7 templates for which the first noun is the head of the subject (table 1), while 4 templates have the second noun as the head of the subject (table 2). The templates were defined after a manual inspection of sentences from the corpus.

We evaluate how the model responds to the generated test inputs. That is, for each test prefix we let the model decide between the singular and plural form of the given verb. We measure the error rate for each template. However, instead of showing the error rates we show how much the language model tends to agree with the most recent noun. This corresponds to the error rate for the templates in table 1, while it corresponds to accuracy for the templates in table 2. Showing the ‘last noun agreement rate’ makes it easier to compare



T1	the _ and the _	0.77
T2	the _ in the _	0.59
T3	the _ by the _	0.69
T4	the _ of the _	0.61
T5	the _ near the _	0.63
T6	the _ at the _	0.70
T7	the _ without the _	0.67

Table 1: Templates for which the number of the verb is opposite to the number of the last noun

the behavior of the model for different templates.

The results are shown in Figure 6, using green and red colors to indicate if the last noun is actually the head of the subject (green) or not (red). We see that all bars are above the 0.5 rate, which shows that the model is most likely to agree with the last noun, even in cases where this is syntactically incorrect. We also see that the red bars are slightly lower than the green bars, 0.67 compared to 0.79 on average. This shows that the model is even more likely to prefer the last noun if this is also syntactically correct. This suggests that the model has at least some sensitivity to syntactic information carried by function words.

We further discuss two special cases, namely the first red bar (T1) and the last green bar (T11). T1, “the \_ and the \_”, is a special case because it actually contains two singular nouns instead of one singular and one plural noun. The predicted verb should be plural because of the conjunction word ‘and’. The existence of two singular nouns makes it hard for the model to pick the plural verb, which could explain the high error rate for T1 in Figure 6. T11 uses different templates for the singular (the \_ ’s \_ ) and the plural (the \_ ’ \_ ) possessive form. The last green bar in Figure 6 shows the average result. We suspect that the relatively low accuracy in this case might be explained by the infrequent occurrence of the plural possessive form in written text. Indeed, a closer inspection of the numbers showed that the singular template had an accuracy of 0.77 (compared to 0.71 for all singular prefixes), while the accuracy of the plural template was lower, namely 0.66 (compared to 0.63 overall for plural prefixes).

We conclude that, although the model performs poorly on syntactically complex sentences it still shows some sensitivity for syntactic information exposed by function words.

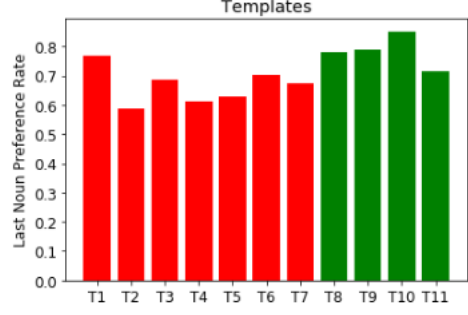


Figure 6: Preference rates for agreement to the last noun. The colors indicate whether this is syntactically correct (green) or incorrect (red).

T8	the _ the _	0.78
T9	the _ that the _	0.79
T10	the _ whether the _	0.85
T11	the _ ’s _ (for plural: the _ ’ _)	0.72

Table 2: Templates for which the number of the verb corresponds to the number of the last noun

#### 4.2.2 Semantic Information

We now look at semantic clues, i.e. companies are more likely to produce than employees. We investigate if the model is able to establish number agreement between the verb and the semantic related noun, ignoring the non-semantic related attractor noun.

1. **TEST DATA:** We construct a testset (100+ sentences) consisting of pairs with singular and plural prefixes, in the following format “The NN of the NNS ...[VBZ/VBP]” and “The NNS of the NN ...[VBZ/VBP]” whereby the head of the subject (the first noun) is semantically related to the verb, while the attractor (the second noun) is randomly picked. An example is: “The prices of the employee ...[stabilizes/stabilize]” and “The price of the employees ...[stabilizes/stabilize]”.

In addition we construct a comparison set consisting of the same prefixes (same verb and attractor noun), but with the difference that the first noun is also randomly picked and therefore most likely not semantically related. Example: “The newspapers of the employee ...[stabilizes/stabilize]” and “The newspaper of the employees ...[stabilizes/stabilize]”.

**EXPERIMENT:** Evaluate our model on the constructed testset and on the comparison set. If the

model scores significantly better on the testset than it shows sensitivity to semantic clues.

#### HOW TO FIND CO-OCCURRENCE RATIOS

We use PMI to measure associations between verbs and nouns.

- find all nouns (NN) and verbs (VBZ) in a tagged corpus (sec02-21.gold.tagged). (pick a random selection if all is too much but make sure you are not only looking at frequent verbs since they tend to be general like 'is/are' instead of specific like 'stabilize')
- build a matrix with nouns as rows and columns as verbs
- fill entries with Count('noun and verb co-occur in a sentence') i.e. loop over all sentences and add one if noun and verb exists in sentence (or use skip grammar, but only count noun that appears before verb, see also lab4)
- Also keep track of the total occurrences for the Verbs and nouns
- replace each entry by the PMI calculated from the co-occurrence count, the total occurrences of the noun, and the occurrences of the verb
- select for each column the entry with the highest score(s) (you may ignore very general verbs like is/are)

Remark: can be learned from tagged data set, no need to look at training data

## 5 Conclusion and Future Work

Overall, we can share the conclusion of (Linzen *et al.*, 2016) that LSTMs are able to capture grammatical structures to some extent. Our model was able to learn to predict the correct number of verb/noun for most of the simplest cases. On more complex sentences, eg. adding intervening nouns or function words in between the subject and the verb it failed to do so. In the latter case we could observe that the model at least shows some sensitivity for syntactic information. Through the further analysis, of the models decisions in the simplest cases, we could find evidence for the assertion that the model just falls back to statistical properties of the training corpus, such as word frequencies and the ratio of plural versus singular words.

For future work it would be interesting to perform the replication of the Linzen experiments on real world sentences rather than on artificially constructed ones. The sentences that we generated might be not semantically meaningful and could be syntactically more divers. It remains open whether our model would perform differently on those.

An interesting aspect of working with LSTMs is to look into the embeddings and further investigate the internal state of the network rather than treating it like a black box. Looking at the activations of the LSTMs might give further insights in the weaknesses of the model and could lead to a better error analysis then by looking solely at the predictions.

To follow up on the experiments it would be also interesting to support the model with explicit syntactic rules in order to generalise instead of relying too much on statistical properties of the training corpus.

## References

- Rafal Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *CoRR*, abs/1602.02410, 2016.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *CoRR*, abs/1611.01368, 2016.
- Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, June 1993.
- Tomas Mikolov, Martin Karafit, Lukas Burget, Jan Cernock, and Sanjeev Khudanpur. Recurrent neural network based language model, 01 2010.
- M Sundermeyer, Ilya Oparin, JL Gauvain, B Freiberg, R Schlter, and Hermann Ney. Comparison of feedforward and recurrent neural network language models, 05 2013.