Empirical Evaluation of LSTM Language Models on Syntactic Dependencies

Maartje de Jonge
Student ID: 194107
maartjedejonge@gmail.com

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

Abstract

LSTM models have become increasingly popular for the task of language modeling, mostly because of their capability to capture long-distance dependencies. Dependencies in natural language are often sensitive to syntactic structure; capturing such dependencies is challenging since LSTM models do not have explicit structural representations. In this paper we focus on noun-verb number agreement as an example of a syntactic dependency. We investigate the sensitivity of LSTM language models to statistic, syntactic and semantic information when predicting the number of the next verb. Our results show that these models are able to learn the numbers for most nouns and verbs using statistic regularities. We also found evidence for a modest sensitivity to syntactic structure exposed by function words.

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. (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 a more explicit grammatical target 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[1] 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. That is, we identified verbs with a strong prediction bias towards either the plural or the singular form, and found a positive correlation with the frequency ratio between the plural and singular forms of those verbs in the training data.

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. The function

[1] Unfortunately the experiment was not completed because one of the team members quit the project.

words 'that' and 'of' provide the crucial information regarding the plurality of the verb. We experimented with various syntactic templates to see if our model shows sensitivity to syntactic structure. The results suggest a modest sensitivity to syntactic information exposed by function words.

The main contribution of this paper is that it provides insights into what information LSTM language models actually use to determine the number of a predicted verb.

Outline The remainder of this paper is organized as follows: We first summarize (Linzen *et al.*, 2016) in Section 2. Next, in Section 3, we replicate some of the experiments of this paper. We then describe our own experiments (Section 4) and end with our conclusion and recommendations for future work (Section 5).

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 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 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 (0.8% for the best performing model) 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 models are sensitive to structural dependencies. To find additional support for syntactic sensitivity the authors of the paper inspected the inner workings of one of these models. 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

In this section we replicate some experiments of (Linzen *et al.*, 2016) to get a general impression of how well our model is able to establish number agreement. All experiments in this paper are performed using an LSTM model[2] trained on a general language modeling task.

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,

[2] https://github.com/pytorch/examples/tree/master/word_language_model

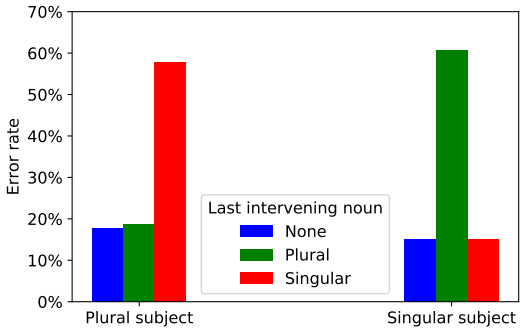

Figure 1: Error rates of the language model plotted against: presence and number of last intervening noun

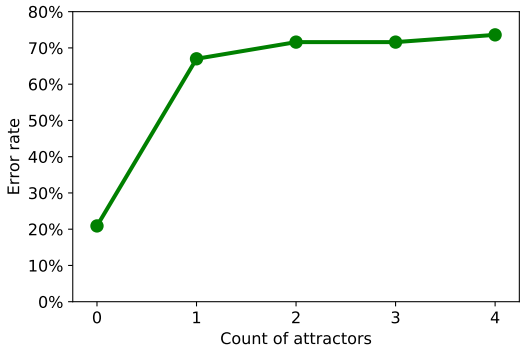

Figure 2: Error rates of the language model plotted against count of attractors in dependencies with homogeneous intervention.

that were amongst the most common and occur in the test corpus as well as in the 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 sentences containing both, the plural (1) and the singular (2) mode of the verb:

the producer plan (1)

the producer **plans** (2)

Following the experiments in (Linzen *et al.*, 2016) we examine the models error rate predicting the number of a verb. That is, the model receives the words leading up to the verb and needs to decide between the singular and plural forms of a particular verb.

We first 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.

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. However, 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 being distracted by an intervening noun with the same number as the subject we only insert nouns with the same number. Linzen et al. (Linzen *et al.*, 2016) refer to such as *dependencies with homogeneous intervention*. Consider the following example sentences:

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

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

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.

Figure 3.2 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 worse than randomly guessing the number of the verb.

4 Own Experiments

In order to predict the correct number of a given verb, the language model should first identify the noun that is the head of the subject for the verb, then decide on the number of that noun, and finally decide on the number of the given verb forms and establish agreement. The latter two are non-trivial since the model has no knowledge of morphological features such as the 's' as a typical postfix for plural nouns and singular verbs.

In the first subsection we investigate if our model is able to do this for simple cases with only a single noun in the prefix. In the second subsection 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 [noun] [verb]". From this we hope to gain insights into whether or not the model is able to learn the plurality number for nouns and verbs. This is a prerequisite for establishing number agreement for simple as well as complex cases.

Using the 40 frequently occuring nouns and verbs that we already used in the replication section, we generated 80 x 40 sentences that cover each noun-verb combination and include both the plural and the singular version of the noun. We tested for each noun-verb combination whether the model predicts the singular or the plural form of the verb, and how sure it is about this prediction.

As a result of this experiment we obtained a matrix in which the rows represent the nouns, the columns represent the verbs, and the entries indicate the models preference for the plural form of the verb. The upper half of the matrix contains all noun-verb combinations for the singular nouns, while the lower half contains the plural nouns. We calculate each entry value as

$$v = \frac{P(< VBP >)}{P(< VBP >) + P(< VBZ >)}$$

where P is the models prediction probability, VBP is the plural form of the verb and VBZ the singular form of the verb. In this manner the matrix entries represent a preference score for plurality ranging from $v = 0$ (surely singular) to $v = 1$ (surely plural). We call v the plurality preference rate.

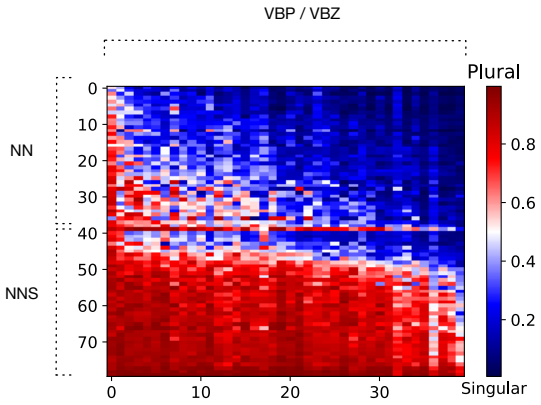

Figure 3: Plurality preference for sentences generated from 40 frequently occuring nouns and verbs. The rows represent the singular (upper half) and plural (lower half) nouns. The columns represent the verbs for which the number must be predicted by the model. A red cell indicates a plural preference for a specific word/noun combination, whereas a blue cell indicates a singular preference.

In order to get a better visual impression we sorted the verbs after their total plurality preference, i.e. the sums of the columns. The rows were sorted in a similar manner, but each half separately in order to conserve the separation between singular (upper half) and plural (lower half) nouns. The resulting matrix is shown in Fig. 3.

As 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 centre as well as at the vertical borders we can observe that there are specific noun-verb combinations for which the model is rather uncertain and/or incorrect. In the following we will investigate the two special cases of verbs and nouns where the model goes wrong.

4.1.1 Verb Statistics

In the columns close to the left border of the matrix, we can find verbs for which the model prefers the plural verb over the singular verb, regardless of the number of the subject noun. For example the verb 'buy' in column 0 has a plurality preference rate of 0.95. At the other hand, in the far right columns we find the verbs for which the singular verb is most often preferred over the plural form. The verb 'say' in column 39, for example, has a

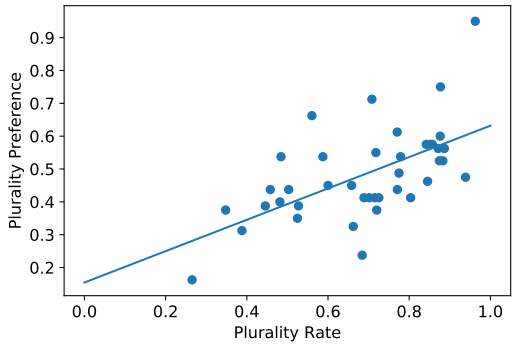

Figure 4: Plurality preference rate of the model plotted against the plurality ratio in the training data, for all verbs in the testset.

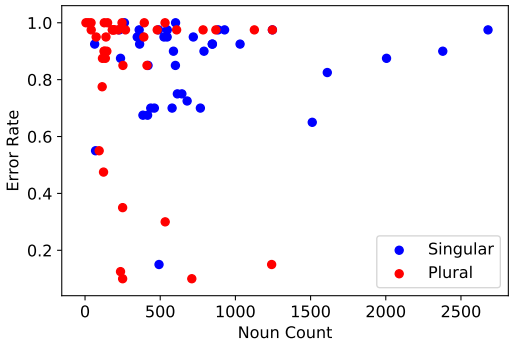

Figure 5: Influence of the noun frequency on the error rate of the model.

plurality preference value of 0.16.

We expect as a possible explanation that for these verbs a high bias exists in the training data for respectively the plural or the singular form. Figure 4 provides some support for this assumption. It shows the plurality rate in the training data plotted against the plurality preference rate for each verb. The plurality rate is measured as

$$r = \frac{Count(< VBP >)}{Count(< VBP >) + Count(< VBZ >)}$$

The line in the figure was generated by performing a linear regression through the points. We see that with an increasing plurality rate the plurality preference rate tends to increase as well. However, not all data points follow this pattern, that is, some data points exist with a high plurality rate and a low plurality preference.

A design descision that we made for this experiment is to count raw verb words without considering their part-of-speech tags. The rationale behind this descicion was that the model also does not have access to part-of-speech tags. However, the consequence of this descision is that we do not distinguish between the plural form (VBP) and the verb base form (VB) when counting the plural verbs. This may explain the overall high plurality rates that we measured in Figure 4. An interesting future experiment could be to preprocess the data with a POS-tagger in order to exclusively count VBP tokens as plural verbs.

4.1.2 Noun Statistics

There are also nouns which show preference for the wrong number and nouns that seem to follow the plurality preference of the verbs, rather than determining the preference with their own plurality number. Around the middle of the matrix we can observe a few nouns that are clearly predicted wrong in most cases. The singular noun ‘income‘ (column 39) with a plurality rate of 0.15 is mostly predicted plural, whereas in columns 40 to 45 the plural nouns ‘weeks‘ (0.9), ‘months‘ (0,9), ‘years‘ (0,85), ‘days‘ (0,7), ‘times’ (0.65), and ‘hours’ (0.53) are incorrectly predicted singular most of the times. Notice that these incorrectly predicted plural nouns are remarkably similar.

To investigate the impact of noun frequencies in the training data on the plurality preference rate of the nouns (and therefore on their error rate), 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. However, as Fig. 5 shows, we could not found clear evidence that a low frequency 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. Furthermore, the frequencies of the mis-counted nouns mentioned before range from 123 to 1241 which is also not exceptional low.

The previous experiment neglects the fact that even though a word is frequent in the corpus it does not neccessarily occur frequently as the subject of a training sentence. For those cases, the model will presumably have more difficulty to establish their numbers correctly; although it could have learned them indirectly from other syntactic properties, for example from the occurences of the plural noun words with a plural count word (two weeks) or from their occurrence with and without a determiner. We manually inspected sentences in

the training data in which the nouns with a high error rate occured. We saw that those nouns were indeed rarely the subject of a sentence. As a future experiment we suggest to use a parser in order to only count the nouns when they act as the subject of a sentence.

4.2 Noun-Verb Agreement in Complex Cases

In Section 3.2 we analysed the performance of the model on complex sentences, containing one or more intervening nouns of opposite number. 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. We investigate if the model uses this information to establish number agreement with the structurally relevant noun, ignoring other irrelevant nouns that occur in a complex sentence.

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 occuring 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 selected, 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 seven templates for which the most recent noun is not the head of the subject (table 1), whereas four templates do have the second noun as the head of the subject (table 2). The templates were defined after 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 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 most recent noun, even in cases where this is

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

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 corresonds to the number of the last noun

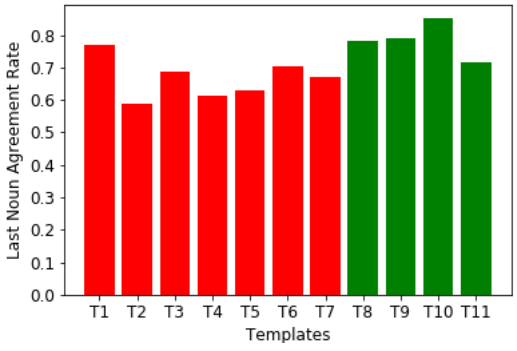

Figure 6: Last noun agreement rates for different templates. The colors indicate whether this is syntactically correct (green) or incorrect (red).

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 a bit less likely to agree with the last noun if this is syntactically incorrect. 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 apparently makes it harder for the model to pick the plural verb, which could explain the high error rate for T1 in Figure 6. The other special case, 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.

4.2.2 Semantic Information

In this section we analyse the models sensitivity to semantic clues, e.g. the price is more likely to stabilize than the products in the sentence "The price of the products ... [stabilizes/stabilize]". The semantic association between the noun 'price' and the verb 'stabilizes' can help the model to identify the word 'price' as the head of the subject, ignoring the semantically non-related attractor noun.

We construct a testset 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 selected. In addition we construct a comparison set consisting of prefixes build from the same verb and attractor noun, but with the difference that the subject noun is also randomly selected. Thus, for the sentences in the comparison set both nouns are most likely to be not semantically associated to the verb. We select strongly associated noun-verb pairs from the corpus, using PPMI[3] as a metric to measure association between nouns and verbs.

We compare the performance of our model on the constructed testset with its performance on the comparison set. If the error rate on the testset is significantly lower than the error rate on the comparison set, then we conclude that the model shows sensitivity to semantic clues[4].

5 Conclusion and Future Work

Overall, we conclude that LSTM language models are able to capture grammatical properties to some extend. Our model was able to predict the correct number of noun/verb pairs for most of the simple sentences; from which we conclude that it does encode the plurality number of those words. Through further analysis of the models decisions in the simplest cases, we could find some evidence that the model occasionally falls back to statistical properties of the training corpus, such as word frequencies. This may happen for example when the ratio between the plural and the singular form of a verb is biased.

On more complex cases, e.g. sentences containing intervening nouns of opposite number in between the subject and the verb, the model most often fails to establish number agreement. In this case we could observe that the model at least shows some sensitivity for syntactic information. That is, the model is most likely to agree with the most recent noun, but it is a bit less likely to do so in case this is syntactically incorrect.

For future work it would be interesting to perform similar experiments on real world sentences rather than on artificially constructed ones. The sentences that we generated are typically not semantically meaningful, in addition, they are syntactically less divers than real world sentences. It remains an open question whether our model would perform differently on those.

An interesting aspect of analysing LSTMs is to look into the embeddings and further investi-

[3] https://en.wikipedia.org/wiki/Pointwise_mutual_information#Applications

[4] Unfortunately the experiment is not completed due to the fact that one of our team members quit the project.

gate 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 into the strenghts and 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 also be interesting to train the model with explicit syntactic structures 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.

Contributions

- **Maartje de Jonge**
  - Abstract
  - Introduction
  - Related Work
  - Own Experiments
    - Simple Cases: Design, Implementation
    - Syntactic Information: Design, Implementation, Text
    - Semantic Information: Design, Text
  - Final editing all sections
- **David Rau**
  - Replication
    - Last Intervening Noun
    - Attractor Counts
  - Own Experiments
    - Simple Cases: Implementation, Text
  - Conclusion and Future Work
  - Implementation Sentence Generator