Do RNNs learn human-like abstract word order: a follow-up study R250 Mini-project Report

Danyi (Oli) Liu
Christ's College
dl567@cam.ac.uk

1 Introduction

In this project[1], I carried out a follow-up study to Futrell and Levy (2018, henceforth F&L) in investigating the capability of RNNs in learning preferences for alternative syntactic constructions involving word orders. F&L showed that RNNs can learn abstract word order preferences involving phrase lengths quite well. To complement their findings, I carried out the same experiment on a synthetic Japanese-like language, where an opposite ordering preference is expected. Despite the limitation of the simplistic training set, the tested sequence model managed to learn to shift long direct object to precede short indirect object in dative construction. I also called into question the use of total sentence surprisal in calculating the preferences for alternative constructions and proposed to use average word surprisals instead.

The report starts by introducing the motivation and relevant work in section 2 and further explained the experiment method in section 3. After detailing the grammar used to generate sentences of the desired synthetic language in section 4, the results of the experiments are presented in section 5. The report concludes with a discussion of the results as well as more broadly about probing RNN syntactic capability using word probability predictions and synthetic languages.

2 Background

Sequence models like RNNs and LSTMs have achieved state-of-the-art performances on many language processing tasks. Although these models perform well in tasks like language modelling and structured prediction, it remains unanswered what these models have actually learnt, and perhaps more importantly, what they cannot learn. In particular, understanding of the hierarchical structure within human language, i.e. syntax, is a crucial component of complete language knowledge, and it is of both scientific interest and practical concern for us to know whether and how sequential, incremental models generalise beyond linear order to abstract structures. In recent years, researchers have explored various ways of understanding the workings of these sequence models, including using formal methods (e.g. Hewitt et al., 2020), surfacing internal representations (e.g. Jumelet et al., 2019), or treating models like psycholinguistic subjects to obtain acceptability judgments from them (e.g. Gulordava et al., 2018).

The work of F&L falls into the last category, where they used surprisal, i.e. the negative log probability of general purpose language models, to reflect the models' preferences between alternative constructions. Specifically, they have focused on syntactic preferences involving abstract word order preferences. They found that RNNs learn soft word orders to a much greater extent than n-gram models, showing particularly strong effect in word order preferences involving constituent length but less evidence for preferences incurred by animacy and definiteness. While English exhibits "short before long" preference, corpus analysis (Dryer, 1980; Hawkins, 1994) and psycholinguistic experiments (Yamashita and Chang, 2001) have shown that Japanese, a SOV, head-final language, shows the opposite trend of shifting long phrases before short ones. The main objective of this follow-up study, therefore, is to test whether RNNs can learn this opposite preference when trained and tested on a Japanese-like artificial language. The success of RNN in learning the abstract word preference in this Japanese-like language would further corroborate the conclusion of F&L, while its failure would provide a starting point for us to further explore the bias and limitation of RNN models.

[1] code available here

3 Methods

3.1 Alternation studied

This project is focused on studying dative alternation, which was investigated in F&L and have empirical support from psycholinguistic experiments (Yamashita and Chang, 2001, henceforth Y&C). In F&L, they have only compared two alternative constructions:

- Double-object (DO) construction:
  The man gave the woman a book.
- Prepositional-object (PO) construction:
  The man gave a book to the woman.

Note that in the later part of this report, DO would denote direct object instead of double-object construction unless explicitly specified. In Y&C, they tested the preferences of human subjects with regard to four competing orders:

- Canonical (SIDV), i.e. subject – indirect object (IO) – direct object (DO) – verb:

  男は 女に 本を あげた。
  Man-wa woman-ni book-wo gave
- IO shift (ISDV), i.e. IO – subject – DO – verb:

  女に 男は 本を あげた。
  woman-ni man-wa book-wo gave
- DO shift (DSIV), i.e. DO – subject – IO – verb:

  本を 男は 女に あげた。
  book-wo man-wa woman-ni gave
- Internal Shift (SDIV), i.e. subject – DO – IO – verb:

  男は 本を 女に あげた。
  man-wa book-wo woman-ni gave

During testing, F&L experimented with all four combinations of theme (their naming for direct object) and recipient (indirect object) lengths: all-short (corresponding to short theme and short recipient), long-IO (long recipient, short theme), long-DO (long theme, short recipient), as well as all-long (long recipient and long theme), while Y&C did not test with all-long sentences in their experiment.

3.2 Measuring RNN judgments

Adopting the approach in F&L, I used surprisal of word w, $S(w) = -\log_2 \mathrm{p}(w)$, to calculate the preference showed by the model between alternative constructions. While F&L have used only sentence surprisal, which is the sum of constituent word surprisals, throughout their work, I also treied computing average word surprisal when trying to reproduce their results. The reason is that the prepositional-object construction is one word longer since it contains an additional preposition *'to'*, and I hope to take this into account when comparing the surprisals of prepositional-object and direct-object constructions. Sentence-level and word average surprisal are equivalent measures for the tests on our Japanese-like artificial language because the preposition is replaced with post-position case-markings which exist in all alternative constructions.

R&L justified the use of surprisal as the measure of preferences from two perspectives. First, total sentence surprisal is directly related to the objective that language models are optimised for during training and hence also closely relates to model performance. Second, word-by-word surprisal can predict human comprehension difficulty, thus reflecting the naturalness or markedness of the sequence.

3.3 Models tested

I directly used the same LSTM model dubbed "GRNN" in F&L, which was originally introduced in Gulordava et al. (2018). The architecture of the model is optimised for the original English Wikipedia corpus, with two layers of 650 units and dropout rate of 0.2. To tailor the model to the smaller and simpler training corpus that I used, I adapted the batch size from 128 to 256, learning rate from 20 to 1, and epochs from 40 to 10.

4 Generating the desired artificial language

Although I discovered near the end of this project that a Japanese LSTM have been used in a similar 'psycholinguistic' study of RNNs (Futrell et al., 2019), it would not have been possible for me to generate enough Japanese test sentences of high quality in reasonable time. Thus, although testing with RNNs trained on actual Japanese corpus could potentially provide more conclusive results, I have worked with synthetic head-final language

with SOV order and case marking in this study.

4.1 Ways to (not) generate the language

I have spent a considerable amount of time exploring ways of generating the desired language. Ideally, this artificial language would have the syntax and word order of Japanese, substantiated with the vocabulary of English to keep the training and testing process relatively easy while maintaining comparability to F&L. To control for word order and case marking to be the only two variables so as to maximise comparability, the new training and testing sentences should ideally be generated from the original ones. I came up with two ways of doing this.

The first approach made use of the corpus creation pipeline designed by Ravfogel et al. (2019). With the same objective of studying the inductive biase of RNNs using typologically different synthetic languages, they manipulated the word order and case system of a corpus given the dependency parse of each sentence. However, when testing their pipeline on sentences from the Wikipedia corpora, I found that the manipulation process was quite fragile and produced sentences with low felicity, particularly when applied to long sentences. Although they tried to move entire subtrees rooted at subject, object, and predicate verb nodes, their manipulation method proved simplistic for sentences with complex constructions and dependency relationships.

The second method attempted to preserve both idiomatic Japanese word order and resemblance to the lexical distribution in the Wikipedia training corpus to the maximum extent. To do this, each sentence in the training corpus is translated to Japanese. After tokenisation, each Japanese word is then mapped to its English counterpart, with the exception of case-markings and post-positions. While there could indeed be errors occurring and propagating in the translation, tokenisation, and lexical mapping processes, the most severe limit on the feasibility of this approach was the inability to obtain the translations of the 3 million sentences in reasonable time and under budget.

4.2 Generating from PCFGs

Using a PCFG to generate sentences of our desired artificial language would be much more efficient in both time and computation power. Moreover, we would be able to control both the lexicon and the syntax of the training text. This control, while allowing us to focus the training process on the type of construction of interest, sacrifices the resemblance of the training data to both realistic input received by a child and normal training data received by general purpose LSTM language models.

Initially, I started with a grammar consisting of three types of sentences (all-short, long IO, long DO). For each sentence, the subject and indirect object (IO) were sampled from the same set of 30 terminal lexical items and were forced to be different within each sentence. The predicate verb was sampled from a set of 30 dative verbs. The direct object (DO) was sampled from 30 nouns. All of the aforementioned non-terminals (subject, verb, DO, IO) contained one terminal word postfixed with the corresponding case marking. There was no overlap between each type of terminal category (subject/indirect-object, verb, general modifier, IO-specific modifier, DO-specific modifier). To generate long IO/DO, case-marked IO/DO was prefixed with an IO/DO modifier. IO and DO shared a common set of 41 general modifiers, and each had 26 IO/DO-specific modifiers. The modifiers are all of the form *"modifier-DO case-marker modifier-verb"* to simulate relative clause modifiers used in the experiments of both Y&C and F&L. The DOs and verbs of the modifiers were disjoint with DOs and verbs of the matrix clause. For all of all-short or long-IO sentences, the phrase ordering was SIDV. For long-DO sentences, the order was SDIV.

Example of terminal items in the grammar

- subject/indirect-object: *woman, teacher...*
- verb: *gave, handed, threw...*
- direct-object: *report, book, letter...*
- general modifier:
  newspaper-ni reported, house-ni stood...
- IO-specific modifier:
  birdwatching-wo enjoyed, house-wo owned...
- DO-specific modifier:
  word-de contained, science(-null) focused...

Sentence types generated from the grammar, corresponding to different combinations of phrase lengths

- all-short:

  subject-*wa* IO-*ni* DO-*wo* verb.
  e.g. *woman-wa teacher-ni book-wo gave.*
  (The woman gave the teacher a book.)

- long-IO:

  subject-*wa* IO-mod IO-*ni* DO-*wo* verb.
  e.g. *woman-wa birdwatching-wo enjoyed teacher-ni book-wo gave.*
  (The woman gave the teacher who enjoyed birdwatching a book.)

- long-DO:

  subject -*wa* DO-mod DO-*wo* IO-*ni* verb.
  e.g. *woman-wa word-de contained book-wo teacher-ni gave.*
  (The woman gave a book that contained word to the teacher.)

The problem with the training set generated in this manner was that, the lexical items of the modifiers would only occur immediately after the subject, i.e. occupying the third to the fifth position in the sentence, for all of the long-DO and long-IO sentences. Since these terminal items do not occur in any other positions in the sentence, the model easily and naturally learnt the preference to place them on the third to the fifth word, and high surprisal would be incurred when these modifier words occurred in other positions. For instance, for an long-IO sentence, the SDIV order would place the modifier words on the fifth to seventh position, resulting in high surprisal since the model had never seen those words there. In other words, the model would learn spurious correlation that is based on remembering linear order rather than abstracting phrase-level alternation.

To obtain greater variation in the distribution of modifier words while maintaining the simplicity of the grammar and some degree of lexical subcategorisation, I enriched the grammar with two types of sentences.

Additional sentence type

- long-IO extra

  subject-*wa* IO-mod(=DO1-*wo* verb1) IO-*ni* DO2-*wo* verb2.
  e.g. *woman-wa letter-ni handed teacher-ni book-wo gave.*
  (The woman gave the teacher who handed a letter a book.)

- all-long

  subject -*wa* IO-mod IO-*ni* DO-mod DO-*wo* verb.
  e.g. *woman-wa birdwatching-wo enjoyed teacher-ni word-de contained book-wo gave.*
  (The woman gave the teacher who enjoyed birdwatching a book that contained word.)

The IO modifier of "long-IO extra" sentences used lexical items from matrix clause DOs and verbs. The "all-long" sentences made it possible for modifier words to occur later in the sentence, i.e. after subject&IO-mod&IO, as well as right after the first subject. I did not include an additional "long DO extra" type where the modifier of DO uses words from matrix clause DO and verb sets due to subcategorisation considerations, i.e. using matrix verbs in relative clause to modify direct objects, like *'report-wo gave book'*, tend to be much less felicitous than, say, *'report-wo gave teacher'*. Maintaining a certain degree of subcategorisation and hence felicity could potentially help reserve some resemblance to normal training data. There are more ways of enriching the training set and varying location distribution of words that I did not adopt in this experiment. For example, we can add in additional constructions other than dative ones (e.g. subject-postposition-verb), or allow increasing modifier length (modifier modified by a modifier).

The training set consisted of 2.3 million sentences, with 0.3 million all-short sentences and 0.5 million long-IO, long-IO-extra, long-DO, and all-long sentences respectively. The training set was randomly shuffled before being fed into the LSTM. The test set contained 200 all-short sentences and 500 sentences from each of the rest four types.

5 Experimental Results

5.1 Training of Language Model

The LSTM model achieved a validation perplexity of 53.4 on the validation set when trained on the original corpus provided by Gulordava et al. (2018) with 3M sentences and a vocabulary of 5k words. When trained on the generated artificial corpus consisting of 3M sentences and 145 unique tokens, the model converged with a much lower validation perplexity of 5.3. An SRN model of the structure described in Gulordava et al. (2018) achieved the exact same validation accuracy. Al-

though it is suspected that the SRN converged to the identical generalisation as the LSTM, the SRN was not used to evaluate the test sentences due to time limitation. The two LSTM models with the same architecture and different training are denoted with the corpus they were trained on from here onward (i.e. Wikipedia LSTM and synthetic LSTM).

5.2 Replicating Futrell & Levy (2018)

As a sanity check, I first tried to replicate the results from F&L using the Wikipedia LSTM. As can be seen in Figure 1 and 2, the LSTM that I trained yielded preferences for prepositional-object construction, as measured by the difference of total sentence surprisal, quite closely to those reported by F&L.

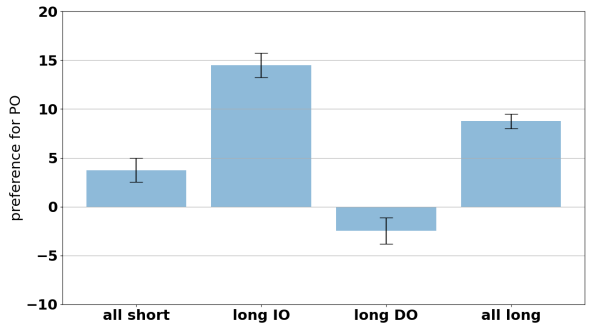

Figure 1: Difference in total sentence surprisal reported by F&L

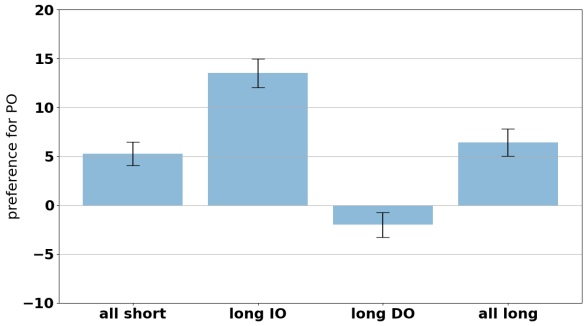

Figure 2: Difference in total sentence surprisal replicated by me

However, if we were to use average word surprisal as the proxy for construction preference, we would find a different pattern in preferences. Using the preference in all-short sentences as the baseline for comparison, the new measurement of preference for prepositional-object would still drop when the DO is long, although not to negative

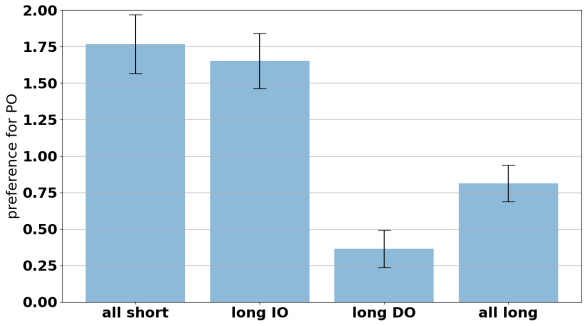

Figure 3: Difference in average word surprisal obtained by me

values. However, having a long IO would not further encourage shifting the IO behind to follow the DO to form a prepositional-object construction. Also, all-long sentences also had a weaker preference for prepositional-object construction than all-short.

Preferences calculated with surprisals in these two slightly different manners exhibited different patterns across the four combinations of IO and DO lengths. This observation raised a question: which preference measure is more appropriate? Although total sentence surprisal is indeed equivalent to the sentence's contribution to the training objective being optimised for, there is little motivation for us to use the difference of sentence surprisal to represent preference between constructions during testing. Although the difference in vocabulary between prepositional-object and direct-object constructions is only one single preposition, the contribution it made to the total sentence surprisal could potentially lead to an overestimation of the surprisal of prepositional-object constructions and hence an underestimation of the preference for prepositional-object constructions.

5.3 Testing on generated language

For the artificial language that I generated, alternative constructions share the exact same vocabulary and thus the total sentence surprisal was just average word surprisal multiplied by sentence length, which was constant for the same combination of phrase lengths (sentence type). Here I reported average word surprisal due to magnitude considerations. When calculating the total sentence surprisal given by the synthetic LSTM, I realised that the difference in sentence surprisal between alternative constructions could be as great as 50, which

is much higher than the difference of sentence surprisal yielded by Wikipedia LSTM. To find out the source of this difference, I computed the average word surprisal across the phrase length combination of long-IO and long-DO within each ordering/constructions.

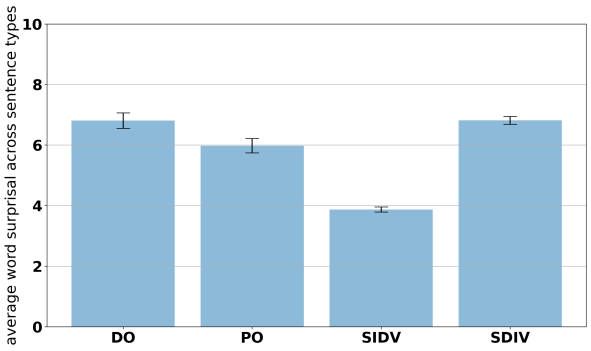

Figure 4: Word surprisal averaged across long-IO and long-DO sentences, grouped by constructions (Direct-Object and Prepositional-Object correspond to average word surprisal given to test sentences by the LSTM trained on the original Wikipedia corpus; SIDV and SDIV refers to that calculated with the LSTM trained on the synthetic corpus)

Although the synthetic LSTM had a much smaller vocabulary (145) than the wikipedia LSTM (5k), the average word surprisal that synthetic LSTM yielded when evaluated on sentences of SDIV ordering, i.e. the ordering of the synthetic language analogous to the prepositional-object (PO) construction in English, was similar to the average word surprisal yielded by the wikipedia LSTM. On the other hand, the average word surprisal for SIDV ordering was much lower. This caused the high-magnitude difference between total sentence surprisal of alternative constructions given by the synthetic LSTM. In other words, the synthetic model have an overall stronger preference for SIDV ordering. This might be related to the spurious correlation mentioned in section 4.2, which was not completely eradicated even after "long-IO extra" and "all-long" sentences were added to vary the constituent words and the location distribution of the modifier.

Despite the synthetic LSTN's overall preference for SDIV ordering, we can see in Figure 5 that the preference for SIDV over SDIV was strongly suppressed when DO is long, suggesting that the synthetic LSTM has learnt to shift the long DO before the short IO.

To evaluate the behaviour of the synthetic

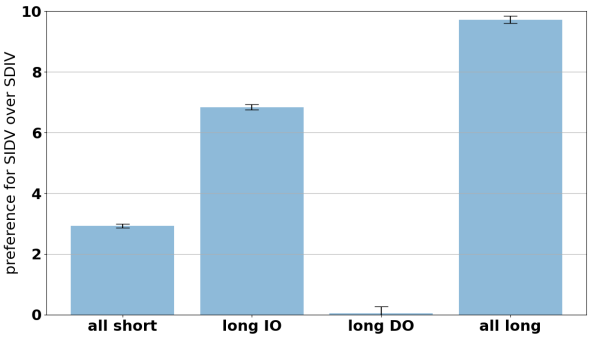

Figure 5: Average word surprisal for SDIV ordering minus SIDV ordering

LSTM against preferences of human subjects recorded in the study of Y&C, I plotted Figure 7 for comparison with Figure 6 taken from Y&C.

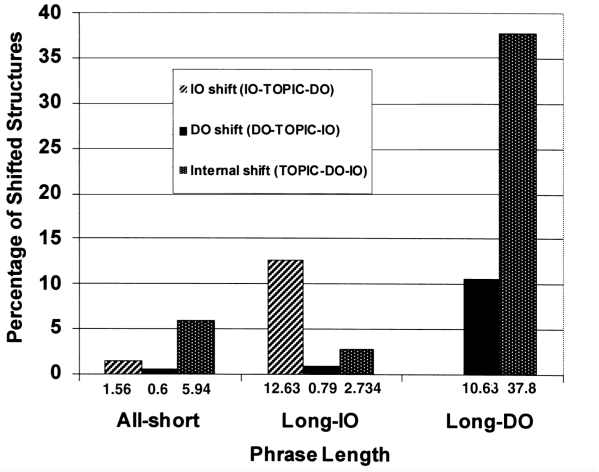

Figure 6: Shifting in dative sentences as a function of phrase length and shift type; with no IO shifts in the Long-DO condition (taken unchanged from Yamashita and Chang (2001).

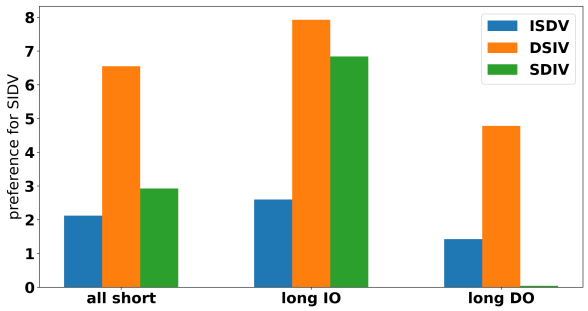

Figure 7: Preference for SIDV over ISDV, DSIV, and SDIV, grouped by combination of phrase lengths

In Figure 6, the height of the bars can be understood as the preferences for shifted orderings (ISDV/DSIV/SDIV) over the canonical or-

der (SIDV) in different combinations of phrase lengths, while the bar heights in Figure 7 reflected the synthetic LSTM's preference for SIDV over alternative orderings. The bars were arranged in the same order, i.e. bars at the same location in the two figures correspond to the same phrase ordering and length combination. Thus, the bar heights in the two figures should correlate negatively if the synthetic were to closely capture the human word preference.

The most prominent observation is again that the LSTM captured the preference for SDIV, i.e. internal shift, when DO is long. As reported by Y&C, internal shift was least preferred by human in long-IO cases, which was also reflected by the LSTM. Another interesting observation from comparing these two figures is perhaps the preference for DSIV ordering. Human subject were much more likely to front DO to the beginning of the sentence when DO was long, which was also exhibited in the synthetic LSTM. As for the fronting of IO (ISDV), while human subjects showed considerable preference in long-IO cases and dis-preference in long-DO cases, the preference for SIDV over ISDV remained consistently small across different phrase length combinations. Admittedly, we could not expect strict negative correlation even if LSTM could perfectly capture human-like ordering preferences, since the synthetic language is hugely simplified and only resembles Japanese to a minimal extent. The test sentences in this study are also not controlled to match those in Y&C. Still, although the negative correlation we observed between Figure 6 and 7 was not consistent across phrase ordering and length combinations, the clear agreement in the internal shifting of DO in long-DO cases was interesting and begs for further investigation.

6 Discussion

In fulfilling the objective to complement F&L, this study provides preliminary evidence of the capability of LSTMs to learn the "long before short" order when working with Japanese-like language. An additional contribution of this study is the proposal of replacing the use of total sentence surprisal with average word surprisal when measuring preferences among alternative constructions, or at least a careful evaluation and comparison of the effect of using these two different measures.

Using the word probability predicted by RNNs as a proxy for acceptability or preferences has become a popular approach in probing the syntactic capability of these sequence models. My worries surrounding this approach is rooted in the variability of the lexical items used in testing. In other words, it is hard to isolate the model's knowledge in syntax and lexical semantics. For example, how do we make sure the model is *really* learning syntax instead of depending on word co-occurrence patterns. In my opinion, comparing the performance of LSTM against that of vanilla n-gram models, as was donw in F&L, is far from enough. The nonce sentences used in Gulordava et al. (2018) is much more convincing, but did not seem to gain much popularity in later work. Also, using different words during testing could potentially result in different results. Averaging across many test sentences could boost confidence in the conclusions drawn from the tests, but it is difficult to judge how many sentences are enough.

Using artificial language could potentially help us control the lexicon with higher precision, and would also allow us to further explore the inductive biases of RNNs across typologically-diverse languages. However, it remains challenging to automatically synthesise the desired artificial language. Methods for manipulating existing corpora to create sentences with desired typological properties are generally computationally expensive, while directly generating artificial languages from a PCFG would potentially require a complex grammar to target at the syntactic phenomenon of interest and at the same time create intricate dependence that goes beyond spurious correlation and reflect the complexity of real languages to a reasonable degree.

Acknowledgement The language models used in my experiments were trained on a departmental GPU machines. Many thanks to Professor Ted Briscoe for his advice.

References

Matthew Synge Dryer. 1980. The positional tendencies of sentential noun phrases in universal grammar. *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 25(2):123–196.

Richard Futrell and Roger P. Levy. 2018. Do RNNs learn human-like abstract word order preferences? *arXiv:1811.01866 [cs]*.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy.

2019. Neural Language Models as Psycholinguistic Subjects: Representations of Syntactic State. *arXiv:1903.03260 [cs]*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless Green Recurrent Networks Dream Hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

John A Hawkins. 1994. *A performance theory of order and constituency*. 73. Cambridge University Press.

John Hewitt, Michael Hahn, Surya Ganguli, Percy Liang, and Christopher D. Manning. 2020. RNNs can generate bounded hierarchical languages with optimal memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1978–2010, Online. Association for Computational Linguistics.

Jaap Jumelet, Willem Zuidema, and Dieuwke Hupkes. 2019. Analysing Neural Language Models: Contextual Decomposition Reveals Default Reasoning in Number and Gender Assignment. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 1–11, Hong Kong, China. Association for Computational Linguistics.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. Studying the Inductive Biases of RNNs with Synthetic Variations of Natural Languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3532–3542, Minneapolis, Minnesota. Association for Computational Linguistics.

Hiroko Yamashita and Franklin Chang. 2001. "long before short" preference in the production of a head-final language. page 11.