Proceedings of the Society for Computation in Linguistics

Volume 5 Article 9

2-1-2022

Learning Argument Structures with Recurrent Neural Network Grammars

Ryo Yoshida
The University of Tokyo, yoshiryo0617@g.ecc.u-tokyo.ac.jp

Yohei Oseki
The University of Tokyo, oseki@g.ecc.u-tokyo.ac.jp

Learning Argument Structures with Recurrent Neural Network Grammars

Ryo Yoshida and **Yohei Oseki**
The University of Tokyo
{yoshiryo0617, oseki}@g.ecc.u-tokyo.ac.jp

Abstract

In targeted syntactic evaluations, the syntactic competence of language models (LMs) has been investigated through various syntactic phenomena, among which one of the important domains has been *argument structure*. Argument structures in head-initial languages have been exclusively tested in the previous literature, but may be readily predicted from lexical information of verbs, potentially overestimating the syntactic competence of LMs. In this paper, we explore whether argument structures can be learned by LMs in head-final languages, which could be more challenging given that argument structures must be predicted before encountering verbs during incremental sentence processing, so that the relative weight of syntactic information should be heavier than lexical information. Specifically, we examined double accusative constraint and double dative constraint in Japanese with the sequential and hierarchical LMs: *n*-gram model, LSTM, GPT-2, and Recurrent Neural Network Grammar (RNNG). Our results demonstrated that the double accusative constraint is captured by all LMs, whereas the double dative constraint is successfully explained only by the hierarchical model. In addition, we probed incremental sentence processing by LMs through the lens of surprisal, and suggested that the hierarchical model may capture deep semantic roles that verbs assign to arguments, while the sequential models seem to be influenced by surface case alignments. We conclude that the explicit hierarchical bias is essential for LMs to learn argument structures like humans.

1 Introduction

Recently, artificial neural networks have had a great impact on the field of Natural Language Processing. Nevertheless, despite the improvement brought by the neural network, it is an open question what linguistic knowledge neural language models (LMs) can learn from the next word prediction task. One line of research peeking into the neural network "black box" is the targeted syntax evaluations with controlled sentences designed to reveal whether the LMs have learned specific syntactic knowledge consistent with human acceptability judgements. (e.g., Lau et al., 2017). Using this method, previous work has shown that these models successfully learn a variety of syntactic knowledge such as subject-verb number agreement (Linzen et al., 2016; Marvin and Linzen, 2018; Wilcox et al., 2018).

In targeted syntax evaluations, one of the important domains has been *argument structure*. Previous work suggested that neural LMs have the ability to capture argument structures (Kann et al., 2019; Warstadt et al., 2020), but in head-initial languages exclusively tested in the previous literature, argument structures may be predicted from lexical information of verbs, potentially overestimating the syntactic competence of the LMs. In addition, although targeted syntax evaluation to test other linguistic knowledge has confirmed the advantage of syntactic bias (Kuncoro et al., 2018; Wilcox et al., 2019; Futrell et al., 2019), hierarchical models such as Recurrent Neural Network Grammars (RNNGs, Dyer et al., 2016) have not been evaluated for verb argument structures.

In this paper, we will examine the effect of syntactic bias on learning verb argument structures, using more challenging head-final language, Japanese. In Japanese, argument structures must be predicted before encountering verbs during incremental sentence processing, such that the relative weight of syntactic information should be heavier than lexical information. We specifically focus on the double accusative constraint (e.g., Harada, 1975, 1986; Shibatani, 1978; Hiraiwa, 2002, 2010) and the double dative constraint in Japanese. The double accusative constraint prohibits the occur-

Proceedings of the Society for Computation in Linguistics (SCiL) 2022, pages 101-111.
Held on-line February 7-9, 2022

Previous literature	English	Italian	Russian	French	German	Hebrew	Basque	Japanese
Linzen et al. (2016), Marvin and Linzen (2018), Jumelet and Hupkes (2018), Chowdhury and Zamparelli (2018, 2019), Wilcox et al. (2018, 2019), Futrell et al. (2019), Warstadt et al. (2019a,b, 2020), Chaves (2020), Da Costa and Chaves (2020), Hu et al. (2020)	✔							
Gulordava et al. (2018)	✔	✔	✔			✔		
Ravfogel et al. (2018)							✔	
An et al. (2019)	✔			✔				
Mueller et al. (2020)	✔		✔	✔	✔	✔		

Table 1: Summary of the previous literature on targeted syntactic evaluations. Works for English are shown above the horizontal line and works for other European languages are shown below the horizontal line.

rences of two or more NPs marked with the accusative case particle *o* within the same clause, and the double dative constraint is the restriction on the case taken by verbs. We will test these constraints with the sequential and hierarchical LMs, *n*-gram, LSTM, GPT-2 (Radford et al., 2019) and Recurrent Neural Network Grammars (RNNGs). As a result, we demonstrated that the double accusative constraint could be captured by all LMs, whereas the double dative constraint is successfully explained only by the hierarchical model. In addition, we analyzed the phrase-by-phrase surprisal of the LMs, and suggested that the hierarchical model may capture deep semantic roles that verbs assign to arguments, while the sequential models are influenced by surface case alignments. This result suggests that the double accusative constraint, which is a constraint to spell out the surface case, can be solved well by the sequential model, but the double dative constraint, which is a constraint at the level of the deep semantic role that verbs assign to arguments, can be solved well only by the hierarchical model. Taken together, we conclude that the explicit hierarchical bias is essential for LMs to learn the human-like syntactic competence to process argument structures.

Another important contribution of this paper is that, to the best of our knowledge, it was the first attempt to conduct targeted syntax evaluation using Japanese. The goal of natural language processing community is to build a LM having language independent general language processing ability, but so far targeted syntax evaluation has been done mainly for English (above the horizontal line in Table 1) and other European languages (below the horizontal line in Table 1). In order to achieve the goal, it is important to evaluate the syntactic competence of LMs for non-European languages.

2 Methods

To investigate the effect of explicitly modeling hierarchical structures, we train linear LMs and a hierarchical LM. In order to eliminate the effect of the amount of training data, we trained all LMs on the same training data. In addition, we restricted our evaluation to left-to-right LMs corresponding to incremental sentence processing, to make LMs predict the verb argument structure before they see the verb. We used the same model sizes reported in the papers proposing each model (Table 2).

2.1 Language Models

Long Short-Term Memory (LSTM): LSTMs are a sequential model using the recurrent neural network architecture (Hochreiter and Schmidhuber, 1997). We used a 2-layer LSTM with 256 hidden and input dimensions. The implementation by Gulordava et al. (2018) was employed.[1]

GPT-2: GPT-2 is a sequential model using the Transformer architecture (Vaswani et al., 2017). We used the same architecture of GPT-2 small (Radford et al., 2019) with 12 layers and 756 hidden and input dimensions. The implementation by Huggingface's Transformer package (Wolf et al., 2020) was employed.

Recurrent Neural Network Grammar (RNNG): RNNGs are a hierarchical model which explicitly models hierarchical structures (Dyer et al., 2016).

[1] https://github.com/facebookresearch/colorlessgreenRNNs

Language Model	#Layers	#Hidden dimensions	#Input dimensions
LSTM	2	256	256
GPT-2	12	768	768
RNNG	2	256	256

Table 2: Model sizes of neural LMs evaluated in this paper.

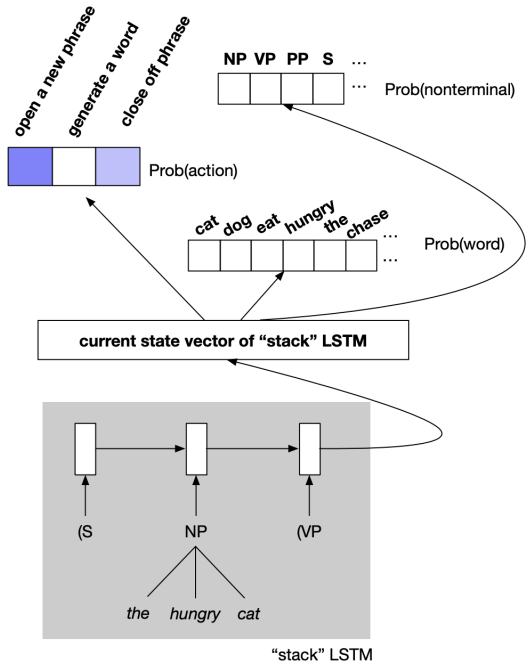


Figure 1: The architecture of RNNGs used in this paper. This figure is reproduced from Hale et al. (2018).

In this paper, we used stack-only RNNGs (Kuncoro et al., 2017). RNNGs generate trees such as "(S (NP *The hungry cat*) (VP *meows*))"; each of the elements is encoded as a vector and stored in a stack, which is illustrated inside the gray box in Figure 1. At each step of generation, one of the following three actions is selected based on the current state of the stack, which is encoded as a vector by stack LSTM:

- **NT(X)** introduces a nonterminal X that is encoded as a vector onto the top of the stack. This action generates an open nonterminal "(X".
- **GEN(x)** introduces a terminal symbol x that is encoded as a vector onto the top of the stack. This action generates a terminal symbol "x".
- **REDUCE** triggers "syntactic composition" function, which creates a new single vector that represents a phrase X from the elements of its children in the stack. For example, "(NP *The hungry cat*)" is represented by a new single vector by this action.

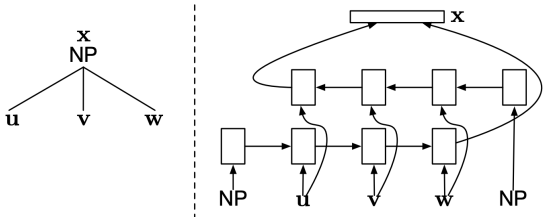


Figure 2: "Syntactic composition" function that is executed during a REDUCE action. This figure is reproduced from Dyer et al. (2016).

If NT(X) or GEN(x) is selected, which open nonterminal or word is generated is selected based on the same vector that represents the current state of the stack.

If REDUCE is selected, "syntactic composition" function is executed by bidirectional LSTM (Figure 2). In both directions, a nonterminal vector such as "(NP" is input first, and then its children vectors such as "**u**", "**v**" and "**w**" are input in forward or reverse order. After all the children vectors are input, the phrase vector "**x**" is calculated from the output of the forward and reverse LSTMs.

We used RNNGs that had a 2-layer stack LSTM with 256 hidden and input dimensions. The implementation by Noji and Oseki (2021) was employed.[2] RNNGs were given the correct tree structures only during training, so we used word-synchronous beam search (Stern et al., 2017) to inference tree structures behind terminal subwords during evaluation. We set the action beam size to 100, the word beam size to 10, and the fast track to 1.

***n*-gram:** As a baseline, we also train 5-gram LM using KenLM.[3]

[2] https://github.com/aistairc/rnng-pytorch

[3] https://github.com/kpu/kenlm

2.2 Training data

All LMs were trained on the National Institute for Japanese Language and Linguistics Parsed Corpus of Modern Japanese (NPCMJ), that comprises 67,018 sentences annotated with tree structures.[4] The sentences were split into subwords by a byte-pair encoding (Sennrich et al., 2016).[5] LSTM, GPT-2, and *n*-gram used only terminal subwords, while RNNGs used terminal subwords and tree structures. All neural LMs (LSTM, GPT-2, and RNNGs) were given one sentence at a time and were trained for 40 epochs and 3 times with different random seeds.[6]

2.3 Acceptability judgements with LMs

Recently, many efforts have been made on the evaluation of the syntactic competence of LMs. Previous work (e.g., Linzen et al., 2016; Marvin and Linzen, 2018; Wilcox et al., 2018) evaluated whether LMs assign a higher probability to an acceptable sentence than to an unacceptable one, using a minimal pair such as (1).

(1) a. The hungry cat meows.

b. *The hungry cat meow.

There are several methods to measure an LM's preference between two sentences in a minimal pair. One of them is prediction task, which compares a probability of grammatically critical position. For example, in the example in (1), we would expect the model to predict $p(\text{meows}|\text{The hungry cat}) > p(\text{meow}|\text{The hungry cat})$. However, the prediction task setting is not applicable when grammaticality is determined by the interaction of several words or when the information necessary to determine grammaticality does not appear in the left context. In this paper, we use the more general full-sentence setting (Marvin and Linzen, 2018; Warstadt et al., 2020), which compares a probability of the two complete sentences. For example, in the example in (1), we would expect the model to predict $p(\text{The hungry cat meows}) > p(\text{The hungry cat meow})$.

[4] http://npcmj.ninjal.ac.jp

[5] Implemented in sentencepiece (Kudo and Richardson, 2018). We set character coverage to 0.9995, and vocabulary size to 8000

[6] Traces and semantic information were removed in the way described in Manning and Schutze (1999).

3 Targeted argument structures

3.1 Double accusative constraint

Japanese has a constraint that prohibits the occurrences of two or more NPs marked with the accusative case particle *o* in the same clause. This constraint is called "double accusative constraint", and have attracted considerable interest in the study of Japanese syntax (e.g., Harada, 1975, 1986; Shibatani, 1978; Hiraiwa, 2002, 2010). One example of double accusative constraint is given in (2):

(2) a. Ken-ga Naomi-**ni/o** gakko-ni
Ken-Nom Naomi-Dat/Acc school-Dat
ik-ase-ta
go-Caus-Past
'Ken made Naomi go to school.'

b. Ken-ga Naomi-**ni** sono-hon-**o**
Ken-Nom Naomi-Dat Dem-book-Acc
yom-ase-ta
read-Caus-Past
'Ken made Naomi read the book.'

c. *Ken-ga Naomi-**o** sono-hon-**o**
Ken-Nom Naomi-Acc Dem-book-Acc
yom-ase-ta
read-Caus-Past
'Ken made Naomi read the book.'

As shown in (2a), when the object NP is marked with the dative case particle *ni*, the causee NP can be marked with either the dative case particle *ni* or the accusative case particle *o*. However, as shown in (2bc), when the object NP is marked with the accusative case particle *o*, the causee NP cannot be marked with the accusative case particle *o* (2c), but must be marked with the dative case particle *ni* (2b).

We can assess the syntactic competence of LMs on double accusative constraint by examining whether LMs assign a higher probability to (2b) than (2c), where both arguments are marked with the accusative case particle *o* within the same sentence. For this purpose, 22 minimal pairs of the (2bc) pattern made by Tamaoka et al. (2018) were collected and the probabilities of the two sentences were compared for each minimal pair. We confirmed that case markers are tokenized into individual subword tokens.

3.2 Double dative constraint

Now we turn to another phenomenon on argument structures: double dative constraint. In Japanese,

Language Model	Accuracy (%)
n-gram	72.7
LSTM	97.0 (± 2.1)
GPT-2	95.5 (± 3.7)
RNNG	**98.5** (± 2.1)

Table 3: The result of targeted syntactic evaluation on double accusative constraint. Average accuracies with standard deviations across different random seeds are reported.

Language Model	Accuracy (%)
n-gram	81.8
LSTM	89.4 (± 8.6)
GPT-2	86.4 (± 3.7)
RNNG	**100.0** (± 0.0)

Table 4: The result of targeted syntactic evaluation on double dative constraint. Average accuracies with standard deviations across different random seeds are reported.

case is marked with particles, and different verbs can take different case patterns. One example of double dative constraint is given in (3):

(3) a. Ken-ga Naomi-**o** gakko-**ni**
Ken-Nom Naomi-Acc school-Dat
oku-tta
take-Past
'Ken took Naomi to school.'

b. *Ken-ga Naomi-**ni** gakko-**ni**
Ken-Nom Naomi-Dat school-Dat
oku-tta
take-Past
'Ken took Naomi to school.'

As shown in (3a), double object verbs take three arguments: an NP marked with the nominative case particle *ga*, an NP marked with the accusative case particle *o*, and an NP marked with the dative case particle *ni*. Double object verbs cannot take an NP marked with the dative case instead of an NP marked with the accusative case (3b), resulting in unacceptable sentences.

We can assess the syntactic competence of LMs on double dative constraint by examining whether LMs assign a higher probability to (3a) than (3b). In order to make the results comparable to the double accusative constraint in the previous section, we contrast (3a) with (3b), where both arguments are marked with the dative case particle *ni* within the same sentence. For this purpose, 22 minimal pairs of the (3ab) pattern made by Tamaoka et al. (2018) were collected and the probabilities of the two sentences were compared for each minimal pair. We confirmed that case markers are tokenized into individual subword tokens.

4 Results

4.1 Double accusative constraint

The result of targeted syntactic evaluation on double accusative constraint is shown in Table 3. Average accuracies with standard deviations across different random seeds are reported. First, the baseline *n*-gram model underperformed the neural LMs. This result demonstrates that the dataset used to test the double accusative constraint cannot merely be solved with local information.

Second, among the neural LMs, the hierarchical model (RNNG) achieved the highest accuracy, while the sequential models (LSTM and GPT-2) also reached the near perfect performance. This result provide evidence supporting that the neural LMs can capture the double accusative constraint without explicitly modeling hierarchical structures.

4.2 Double dative constraint

The result of targeted syntactic evaluation on double dative constraint is shown in Table 4. Average accuracies with standard deviations across different random seeds are reported. First, the baseline *n*-gram model performed relatively well, but the performance is still lower than the neural LMs. This result indicates that the dataset used to test the double dative constraint can reasonably be solved with local information alone, but neural architectures may be required to reach the higher performance.

Second, among the neural LMs, the hierarchical model (RNNG) achieved the perfect accuracy, whereas the sequential models (LSTM and GPT-2) did not reach the near perfect performance with only slight improvements over the baseline *n*-gram model. This result provide evidence supporting that, unlike the double accusative constraint, the neural LMs can capture the double dative constraint only when explicitly modeling hierarchical structures.

5 Probing sentence processing

Sections 3.1 and 3.2 demonstrated that the explicit hierarchical bias may not be necessary for the double accusative constraint, but crucial for the double

dative constraint. Why can the sequential models learn the double accusative constraint, but not the double dative constraint? In this section, following Futrell et al. (2019), we probe sentence processing and identify the phrases where LMs make different predictions for acceptable and unacceptable sentences by computing phrase-by-phrase surprisal of LMs. The following analyses include neural LMs to the exclusion of the n-gram model.

5.1 Methods

We probed sentence processing of LMs through the information-theoretic complexity metric called *surprisal* (Hale, 2001; Levy, 2008): $-\log p(\text{segment}|\text{context})$. In psycholinguistics, it is well known that humans predict next segments during incremental sentence processing, and the less predictable the segment is, the more surprising that segment is. The previous literature established that cognitive efforts measured from humans are proportional to surprisals computed from LMs (e.g., Smith and Levy, 2013; Frank and Bod, 2011; Frank et al., 2015). Building on this result, we probe sentence processing by LMs through the lens of surprisal.

5.2 Results

5.2.1 Double accusative constraint

Figure 3 shows phrase-by-phrase surprisal for the double accusative constraint. Phrasal surprisal was computed as the cumulative sum of surprisals of its constituent subwords. Average surprisals with standard errors across different items and random seeds are reported.

We observe that all LMs show the largest surprisal difference at the accusative case particle *o* marking the third NP. This observation suggests that the all LMs captured the double accusative constraint through consecutive case marking on the second and third NPs. Notice incidentally that only RNNG shows larger surprisal at the end of unacceptable sentences than acceptable sentences.

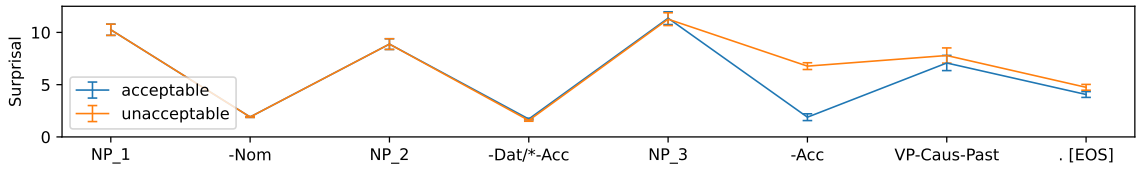
5.2.2 Double dative constraint

Figure 4 shows phrase-by-phrase surprisal for the double dative constraint. Phrasal surprisal was computed as the cumulative sum of surprisals of its constituent subwords. Average surprisals with standartd errors across different items and random seeds are reported.

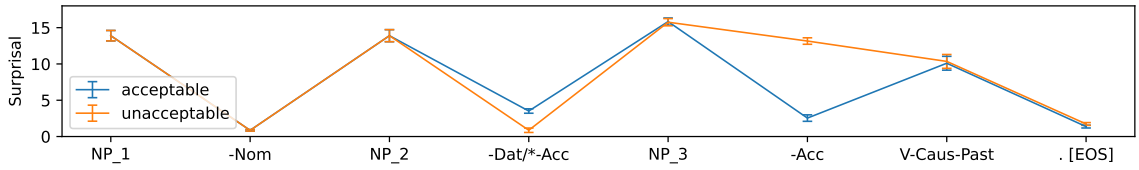
First, unlike the double accusative constraint, we cannot observe the phrases where all LMs consistently show a large surprisal difference. Second, LSTM and GPT-2 show the largest surprisal difference at the dative case particle *o* marking the third NP, while RNNG shows the largest surprisal difference at the case particle marking the second NP. These observations suggest that the sequential models are more surprised when the dative case particle *ni* marks two NPs consecutively, while the hierarchical model is more surprised when the dative case particle *ni* marks the second NP incorrectly, which should be marked by the accusative case particle *o*.

In order to confirm this result, we statistically tested via paired-samples *t*-tests whether the "surprisal differences between acceptable and unacceptable sentences" are significantly different between the case particle marking the second NP (the phrase where the dative case particle marks one NP incorrectly) and the case particle marking the third NP (the phrase where the dative case particle marks two NPs consecutively). The result revealed that LSTM shows a significantly larger surprisal difference at the case particle marking the third NP ($p < 0.05$), while RNNG shows a significantly larger surprisal difference at the case particle marking the second NP ($p < 0.05$), but GPT-2 did not show any significant difference ($p = 0.067$). In other words, LSTM was more surprised when the dative case particle *ni* marks two NPs consecutively, while RNNGs were more surprised when the dative case particle *ni* marks the second NP incorrectly, which should be marked by the accusative case particle *o*, but GPT-2 was equally surprised at both phrases. The important conclusion here is that the hierarchical model (RNNG) not only achieved the perfect accuracy but also captured the double dative constraint for right reasons (i.e. incorrect case marking on the second NP), while the sequential models (LSTM and GPT-2) solved the double dative constraint for wrong reasons (i.e. consecutive case marking on the second and third NPs). In fact, as in (4), it is possible to have consecutive dative cases in Japanese, for example, when a NP marked by the dative case expresses time, and it is wrong to judge ungrammaticality on the basis of a series of dative cases.

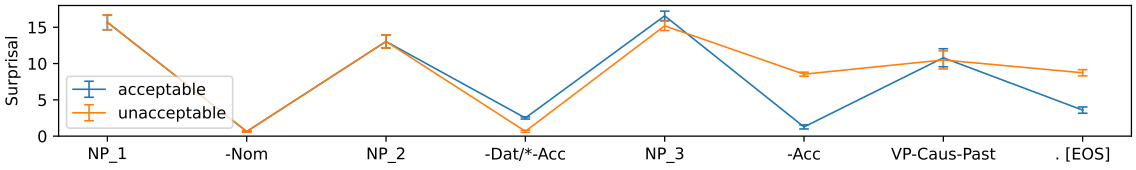
(4) Ken-ga yoake-**ni** gakko-**ni** i-tta
Ken-Nom dawn-Dat school-Dat go-Past
'Ken went to school at dawn.'



(a) LSTM

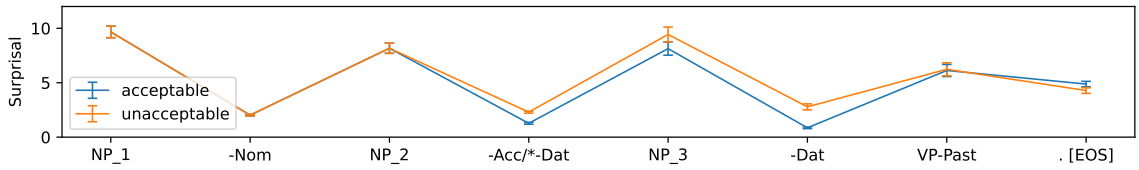


(b) GPT-2

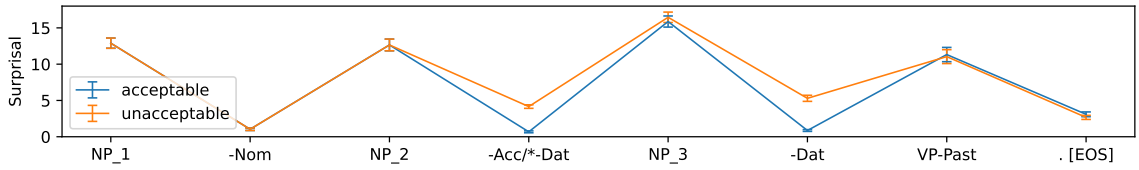


(c) RNNG

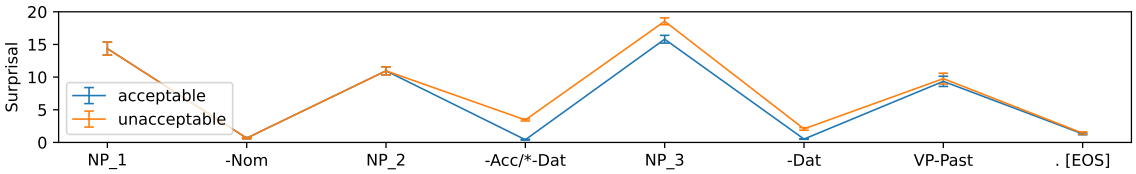
Figure 3: Phrase-by-phrase surprisal for the double accusative constraint. Phrasal surprisal was computed as the cumulative sum of surprisals of its constituent subwords. Average surprisals with standard errors across different items and random seeds are reported.



(a) LSTM



(b) GPT-2



(c) RNNG

Figure 4: Phrase-by-phrase surprisal for the double dative constraint. Phrasal surprisal was computed as the cumulative sum of surprisals of its constituent subwords. Average surprisals with standard errors across different items and random seeds are reported.

6 General discussion

In summary, we demonstrated that all LMs can capture the double accusative constraint, while only the hierarchical model can solve the double dative constraint with the perfect accuracy. Moreover, further analyses of incremental sentence processing revealed that the double accusative constraint can be attributed to the phrase where the second and third NPs are marked consecutively, while the double dative constraint seems to be adequately captured by the hierarchical model at the phrase where the second NP is marked incorrectly. In this section, we discuss these results from the perspective of theoretical linguistics.

First, Hiraiwa (2010) proposes that the double accusative constraint is not a pure syntactic constraint, but an interface constraint on the spell-out of the accusative case; namely, the phonological constraint against realizing multiple occurrences of the accusative case value within the same domain. Interestingly, this proposal is consistent with our results in that the double accusative constraint is modeled by LMs through surface case alignments like consecutive case marking on the second and third NPs.

Second, the double dative constraint, on the other hand, seems to be a pure syntactic constraint, where NPs should be marked with the accusative case particle given deep semantic roles (i.e. theme) that verbs assign to arguments. Among the neural LMs tested above, only RNNG distinguished unacceptable sentences from acceptable sentences at the phrase where the second NP is marked incorrectly. Although GPT-2 also shows a similar trend to RNNG, the sequential models seem to be surprised for wrong reasons by consecutive case marking on the second and third NPs, which is not the critical point of the difference between acceptable and unacceptable sentences. This result may suggest that the sequential models cannot learn deep semantic roles that verbs assign to arguments and, alternatively, are strongly influenced by surface heuristics (McCoy et al., 2019). In contrast, the hierarchical model can learn those deep semantic roles by explicitly modeling hierarchical structures (Wilcox et al., 2020).

7 Limitations and future work

In this paper, we performed the targeted syntactic evaluation of LMs on argument structure in Japanese, which could be more challenging than English given that argument structures must be predicted before encountering verbs during incremental sentence processing. However, our results suggests that the dataset used in this paper may be too easy: even the baseline n-gram model can solve well (accuracy = 72.7% on double accusative constraint and 81.8% on double dative constraint). We should evaluate LMs on more challenging dataset to strengthen the argument in this paper.

In addition, in order to make the fair comparison of different architectures of the LMs, we trained all LMs on NPCMJ, the largest treebank in Japanese. However, since NPCMJ is relatively small (67,000 sentences), and the previous literature has shown that sequential models can reach the higher performance comparable to hierarchical models when trained on larger training data (Futrell et al., 2019), whether the results scale or not remains to be explored in future work.

Finally, this paper was the first attempt to conduct the targeted evaluation in Japanese, but only two syntactic phenomena on argument structures were examined in this paper. In order to scale the targeted syntactic evaluation, we plan to evaluate the syntactic competence of LMs on a wider range of syntactic phenomena in Japanese. We hope that this paper will motivate the targeted evaluation of the syntactic competence of LMs across languages.

8 Conclusion

In this paper, we explored whether argument structures can be learned by LMs in head-final languages, where argument structures must be predicted even before encountering following verbs during incremental sentence processing. Specifically, we examined double accusative constraint and double dative constraint in Japanese with the sequential and hierarchical LMs: n-gram model, LSTM, GPT-2, and RNNG. Our results demonstrated that the double accusative constraint could be captured by all LMs, whereas the double dative constraint is successfully explained only by the hierarchical model. In addition, we probed sentence processing by LMs through the lens of surprisal, and suggested that the hierarchical model may capture deep semantic roles that verbs assign to arguments, while the sequential models are influenced by surface case alignments. We conclude that the explicit hierarchical bias is essential for LMs to learn the human-like syntactic competence to process argument structures.

Acknowledgements

We would like to thank three anonymous reviewers of the *Society for Computation in Linguistics* for valuable comments and suggestions. This work was supported by JST PRESTO Grant Number JPMJPR21C2, and developed from a term paper submitted to the graduate course titled "Foundations of Linguistic Analysis I" offered at the Department of Language and Information Sciences, Graduate School of Arts and Sciences, University of Tokyo in Spring 2021.

References

Aixiu An, Peng Qian, Ethan Wilcox, and Roger Levy. 2019. Representation of constituents in neural language models: Coordination phrase as a case study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2888–2899, Hong Kong, China. Association for Computational Linguistics.

Rui Chaves. 2020. What don't RNN language models learn about filler-gap dependencies? In *Proceedings of the Society for Computation in Linguistics 2020*, pages 1–11, New York, New York. Association for Computational Linguistics.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shammur Absar Chowdhury and Roberto Zamparelli. 2019. An LSTM adaptation study of (un)grammaticality. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 204–212, Florence, Italy. Association for Computational Linguistics.

Jillian Da Costa and Rui Chaves. 2020. Assessing the ability of transformer-based neural models to represent structurally unbounded dependencies. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 12–21, New York, New York. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent Neural Network Grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

Stefan Frank and Rens Bod. 2011. Insensitivity of the Human Sentence-Processing System to Hierarchical Structure. *Psychological science*, 22:829–34.

Stefan L. Frank, Leun J. Otten, Giulia Galli, and Gabriella Vigliocco. 2015. The ERP response to the amount of information conveyed by words in sentences. *Brain and Language*, 140:1–11.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. 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 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

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 Hale. 2001. A Probabilistic Earley Parser as a Psycholinguistic Model. In *Proceedings of the Second Conference of the North American Chapter of the Association for Computational Linguistics*, pages 159–166.

John Hale, Chris Dyer, Adhiguna Kuncoro, and Jonathan Brennan. 2018. Finding syntax in human encephalography with beam search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2727–2736, Melbourne, Australia. Association for Computational Linguistics.

S. I. Harada. 1986. "counter equi-np deletion". *Journal of Japanese Linguistics*, 11(1-2):157–202.

Shin-Ichi Harada. 1975. The functional uniqueness principle. *Attempts in linguistics and literature*, 2:17–24.

Ken Hiraiwa. 2002. Facets of case: On the nature of the double-o constraint. In *The proceedings of the 3rd Tokyo Psycholinguistics Conference (TCP 2002)*, pages 139–163. Citeseer.

Ken Hiraiwa. 2010. Spelling out the double-o constraint. *Natural Language & Linguistic Theory*, 28(3):723–770.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation*, 9(8):1735–80.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Jaap Jumelet and Dieuwke Hupkes. 2018. Do language models understand anything? on the ability of LSTMs to understand negative polarity items. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.

Katharina Kann, Alex Warstadt, Adina Williams, and Samuel R. Bowman. 2019. Verb argument structure alternations in word and sentence embeddings. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, pages 287–297.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. 2017. What Do Recurrent Neural Network Grammars Learn About Syntax? In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1249–1258, Valencia, Spain. Association for Computational Linguistics.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, Acceptability, and Probability: A Probabilistic View of Linguistic Knowledge. *Cognitive Science*, 41(5):1202–1241.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

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

Christopher Manning and Hinrich Schutze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Hiroshi Noji and Yohei Oseki. 2021. Effective batching for recurrent neural network grammars. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4340–4352, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Shauli Ravfogel, Yoav Goldberg, and Francis Tyers. 2018. Can LSTM learn to capture agreement? the case of Basque. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 98–107, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Masayoshi Shibatani. 1978. *Nihongo no bunseki*. Taishukan Publishing Company.

Nathaniel J. Smith and Roger Levy. 2013. The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective Inference for Generative Neural Parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700, Copenhagen, Denmark. Association for Computational Linguistics.

Katsuo Tamaoka, Jingyi Zhang, and Toshiki Satoh. 2018. An experimental study on psychological reality of double accusative constraint by the maze task. *Studia Linguistica*, 32:115–130.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proceedings of NIPS*, pages 5998–6008.

Alex Warstadt, Yu Cao, Ioana Grosu, Wei Peng, Hagen Blix, Yining Nie, Anna Alsop, Shikha Bordia, Haokun Liu, Alicia Parrish, Sheng-Fu Wang, Jason Phang, Anhad Mohananey, Phu Mon Htut, Paloma Jeretic, and Samuel R. Bowman. 2019a. Investigating BERT's knowledge of language: Five analysis methods with NPIs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2877–2887, Hong Kong, China. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: A benchmark of linguistic minimal pairs for English. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 409–410, New York, New York. Association for Computational Linguistics.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019b. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019. Structural supervision improves learning of non-local grammatical dependencies. 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 3302–3312, Minneapolis, Minnesota. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Ryosuke Kohita, Roger Levy, and Miguel Ballesteros. 2020. Structural supervision improves few-shot learning and syntactic generalization in neural language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4640–4652, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.