Target Evaluation for Neural Language Model using Japanese Case Frame

Kazuhito Tamura[1], Ikumi Suzuki[2] and Kazuo Hara[3]
[1]*Graduate School of Science and Engineering, Yamagata University, Yonezawa-shi, Yamagata, Japan*
[2]*School of Information and Data Sciences, Nagasaki University, Nagasaki-shi, Nagasaki, Japan*
[3]*Faculty of Science, Yamagata University, Yamagata-shi, Yamagata, Japan*

Keywords: Neural Language Model, Target Evaluation, Japanese Case Frame, LSTM.

Abstract: Automatic text generation are widely used in various type of natural language processing systems. It is crucial to capture correct grammar for these systems to work. According to the recent studies, neural language models successfully acquire English grammar. However, it's not thoroughly investigated why the neural language models work. Therefore, fine-grained grammatical or syntactic analysis is important to assess neural language models. In this paper, we constructed grammatical evaluation methods to assess Japanese grammatical ability in neural language models by adopting a target evaluation approach. We especially focus on case marker and verb match in Japanese case grammar. In experiments, we report the grammatical ability of neural language model by comparing n-gram models. Neural language model performed better even some information lacks, while n-gram performs poorly. Also, Neural language model exhibited more robust performance for low frequency terms.

1 INTRODUCTION

Many modern natural language processing systems such as, summarization, machine translation, question answering, dialogue systems, etc., employ automatic text generation. Neural language model has become mainstream and fundamental technique in recent years.

For these systems to work, it is important that automatically generated texts follow correct grammar. In recent reports, the neural language model exhibits the ability to capture the correct grammar. In case of English, the neural language model obtained by learning a large amount of English text has succeeded in acquiring English grammar (Gulordava et al., 2018; Marvin and Linzen, 2018).

In evaluating natural language processing systems, perplexity, BLEU, and other systematic approaches are widely applied (Papineni et al., 2002). However, it's desirable to assess the systems with fine-grained grammatical or syntactic analysis to understand performances in detail (Sennrich, 2017).

Target evaluation is one way to evaluate grammatical and syntactic analysis (Marvin and Linzen 2018). Marvin and Linzen (2018) automatically constructed different variations of structure-sensitive grammatical and ungrammatical sentences to evaluate language models and compare the probabilities the models assign to the sentences.

These works provide new informative ways of fine-grained grammatical evaluation on language models (Warstadt et al., 2019; Kim et al., 2019) and further improvement on the language model by using grammatical and ungrammatical sentences for learning (Nochi and Takamura, 2019).

As evaluation methods for Japanese neural language models, BLUE and human evaluation are widely applied (Imamura et al., 2018; Miyazaki and Shimizu, 2016; Hasegawa et al., 2017). However, there is no fine-grained grammatical evaluation method based on Japanese case grammar, to the best of our knowledge.

In this paper, we investigate the ability of the neural language model for Japanese grammar. Especially, we focus on case marker and verb relations in Japanese case grammar. To achieve this goal, firstly we construct evaluation datasets. Secondly, we examine the neural language model of grammatical ability in Japanese by comparing n-gram models.

Tamura, K., Suzuki, I. and Hara, K.
Target Evaluation for Neural Language Model using Japanese Case Frame.
DOI: 10.5220/0010137702510258
In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2020) - Volume 1: KDIR*, pages 251-258
ISBN: 978-989-758-474-9

2 RELATED WORK

2.1 Evaluation on Language Model

Perplexity and target evaluation (Marvin and Linzen, 2018) were the methods for directly evaluating language models. Perplexity represents the complexity of the model. Generally, model performance was judged on how high probability a model can assign to test data. Target evaluation is methods that focuses on the grammatical or syntactic structure of the language, such as the numerical match between a noun and a verb. More detail on target evaluation for neural language models (Marvin and Linzen, 2018) is described in the next section 2.2.

As a method of evaluating the text generated by the language model, human evaluation and BLEU (Papineni et al., 2002) are commonly used methods.

For Japanese evaluation on neural language models, BLEU scores are applied for machine translation tasks (Imamura et al., 2018) and image captioning (Miyazaki and Shimizu, 2016). ROUGE and human evaluation were applied for sentence compression (Hasegawa et al., 2017). Many Japanese evaluation methods for neural language models are depend on BLEU, ROUGE and human evaluation.

2.2 Target Evaluation on Neural Language Model

Target evaluation (Marvin and Linzen, 2018) is a method for evaluating language models. When there is specific type of interest to evaluate, target evaluation is applicable.

For machine translation, Sennrich (2017) created datasets which captures some type of translation error automatically and reproducibly. In modeling semantics, Zweig et al. (2011) created imposter sentences to compare or evaluate semantic systems.

To evaluate performance of language models, Linzen et al. (2016) mainly evaluated the language model to judge whether the noun form and the verb form match. For example, while a grammatical sentence such as "The author laughs" was created, an ungrammatical sentence "The author laugh" was created as a pair of evaluation sentences. Then, it is expected that the higher probability would be assigned by the neural language model to a grammatical sentence rather than an ungrammatical sentence. Also, the effect of distance between noun and verb by comparing likelihoods of each target verbs were evaluated. Gulordava et al. (2018) developed grammatically correct, but semantically nonsensical sentences to purely evaluate syntactic ability of the language models. Marvin and Linzen (2018) proposed to create datasets for grammatical and ungrammatical sentence pair represent different variations of structure-sensitive phenomena.

To evaluate Japanese grammatical ability in neural language model, we propose target evaluation by focusing on the case and verb relation in Japanese case grammar. Before introducing our proposal, we introduce Japanese grammar and Japanese case frame in the following section 2.3 and 2.4.

2.3 Case Grammar

The case grammar analyze structure of sentences based on the case (Fillmore, 1971). In Japanese case grammar, nouns are associated with predicates with a grammatical role. This grammatical role is called "case". Semantic relations of the case are called deep cases, and syntactic relations are called surface cases. While the surface case of English is expressed by word order and preposition, the surface case of Japanese is expressed by case particles (The National Language Research Institute, 1997).

In case frame, case marker determines the case. In Japanese, case particles mainly play the role of case marker.

2.4 Japanese Grammar

Japanese is a head-final language, and the case particle plays the role of case marker. Unlike English, word order does not determine the case (Kawahara and Kurohashi, 2002). In this paper, a case particle is denoted as a case marker.

The structure of Japanese sentence is shown below as an example;

(1)	Kare *ga*	eiga *wo*	miru.
	He	movie	watch

This sentence is composed of two cases, (Kare *ga*) and (eiga *wo*), and a verb (miru). The noun (Kare) is accompanied by the case marker (*ga*). The verb "miru" takes "*ga*" as nominative case marker and "*wo*" as an accusative case marker.

As shown in the above example, a verb co-occur with cases and case marker indicates the case. Therefore, the combination of verb and case marker is important in Japanese grammar. To find out which case marker co-occur with verb, we refer to case frame dictionary.

2.5 Japanese Case Frame

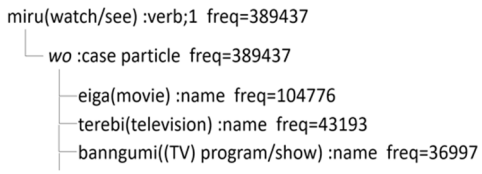


Figure 1: An illustration of case frame dictionary. A verb "miru (see/ watch)" (見る) takes "*wo*" as a case marker . The "wo" case marker appeared with nouns such as, "eiga" 映画, "terebi"(テレビ), etc. in corpus.

To collect verb and co-occur case markers examples, we refer to the Kyoto University Case Frame Dictionary (Kawahara and Kurohashi, 2006a, 2006b). As resource, GSK2008-B Ver 2.0, was obtained from GSK (Language Resources Association in Japan)[1].

The Kyoto University Case Frame Dictionary stores example-based case frames which was constructed from a huge raw corpus. Verbs are classified according to the case usage with examples. For simplicity in this paper, the individual usages of verbs are not distinguished, that is, the cases distinguished by those usage are merged.

For each verb, case markers are listed which co-occurred with the verb in the corpus. We define these case markers as correct set of case markers for the verb. And if case markers which are not co-occurred with the verb are defined as incorrect case markers. Note that the nouns which accompanied with case markers are also recorded in the dictionary.

The case markers we use in this paper are "*ga*", "*wo*", "*ni*", "*de*", "*to*", "*he*", "*kara*", "*made*", "*yori*" and "*no*". Since we focus on case markers, other cases which are not case particles is omitted in this paper.

Figure. 1 shows an example of a case frame for verb "miru (see/watch)". From this example, the verb "miru" takes "*wo*" as a case marker. And "*wo*" case marker is appeared with the nouns such as "eiga", "terebi", etc. in the corpus. The frequency of the verb "miru" (see, watch) is 389,437 and the frequency of the noun "eiga" (movie) is 104,776. This indicates that there are 104,776 sentences in the corpus that have each structure of "eiga wo miru" (watch movie). We will refer these frequencies to construct target evaluation dataset. More details are explained in section 4.

Table1: Case markers occurred with the verbs. This table is summarized from the Kyoto University Case Frame dictionary. For example, the verb "miru" can take "*ga*", "*wo*" as a case marker, but not "*he*". The first column is the list of verbs and the row corresponds to case markers.

Case marker	*ga*	*wo*	*ni*	*he*	*to*	*de*	*kara*	*yori*	*made*	*no*
omou	✓	✓	✓	-	✓	✓	✓	-	-	✓
miru	✓	✓	✓	-	✓	✓	✓	✓	✓	✓
motu	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
sagasu	✓	✓	✓	-	✓	✓	✓	✓	-	✓
tukuru	✓	✓	✓	-	✓	✓	✓	-	✓	✓
tuku	✓	✓	✓	-	✓	✓	✓	-	✓	✓
kangaeru	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
yomu	✓	✓	✓	-	✓	✓	✓	✓	✓	✓
kannziru	✓	✓	✓	-	✓	✓	✓	✓	✓	✓
svoukaisuru	✓	✓	✓	-	✓	✓	✓	-	✓	✓

3 PROPOSED METHOD

Target evaluation is one of the ways to evaluate grammatical and syntactic analysis. Especially, when there is specific type of interest to evaluate, target evaluation is applicable.

In this paper, we propose a target evaluation in Japanese case grammar by assessing "match between case marker and verb". To construct datasets, we generate a pair of a grammatical and an ungrammatical sentence. In target evaluation, the model is evaluated whether the constructed language model can distinguish between a grammatical and ungrammatical sentence.

In this section, we briefly illustrate with a simple example how to generate sentences with an example.

To generate a pair of grammatical and ungrammatical sentence, case markers of verbs play the key role. Table 1 shows summarization of case markers for each verb recorded in The Kyoto University Case Frame Dictionary. By referring to Table 1, we can tell which case markers are included in the correct set of case markers for a verb and which case markers are not. As an example, when a sentence with a verb "miru" (see/ watch) is generated, case markers are selected from the correct set of case markers for the verb. For the verb "miru" (see/ watch), the correct set of case markers includes "*ga*" case, "*wo*" case, "*ni*" case and "*to*" case from Table 1. Then a grammatical sentence for the verb "miru" (see/watch) is generated as follows;

(2) ☐ *ga* ☐ *wo* ☐ *ni* ☐ *to* miru.

[1] https://www.gsk.or.jp/catalog/gsk2008-b/

Each blank □ is filled with an appropriate noun for each case markers. More precise explanation, such as how to choose the nouns and the order of the case, are explained in section 4.

Ungrammatical sentence is generated by replacing an arbitrary case marker in the generated grammatical sentence with a case marker which is not in the correct set of case markers for the verb. Since the sentence contains an incorrect case marker, the sentence is not considered as a correct Japanese sentence. As an example, the verb "miru" (see/watch) does not take the "*he*" case marker as shown in Table 1. Then, arbitrary case marker, such as "*to*" case in the grammatical sentence (2) is replaced with "*he*" case marker to generate an ungrammatical sentence. An example of ungrammatical sentence is as follows;

(3) □ *ga* □ *wo* □ *ni* □ ***he*** miru.

The only difference of sentence (2) and (3), where the sentence (2) use "*to*" case and the sentence (3) use "*he*" case, makes the difference of grammatical and ungrammatical sentence in this example. Then the language models are evaluated whether language model can distinguish between a grammatical and ungrammatical sentence.

4 EXPERIMENTS

In this section, we explain how to construct datasets, the language models (LSTM) and n-grams to compare and evaluation with the datasets for the constructed language models.

4.1 Evaluation Dataset

The purpose of this paper is to evaluate whether language model can distinguish between a grammatical and ungrammatical sentence, that is, match between case marker and verb. We investigate whether language models will change prediction depending on the amount of information before or after the case marker in question. To do so, two type of evaluation datasets are constructed.

Dataset 1. A sentence includes 5 case markers and a verb.

Dataset 2. A sentence includes 1 to 5 case markers and a verb.

To evaluate the effect of distance between case marker and verb, the distance between incorrect case marker and verb varies. When selecting one case marker in a grammatical sentence to generate an ungrammatical sentence, the case marker is selected according to the distance from the verb and replaced with incorrect one.

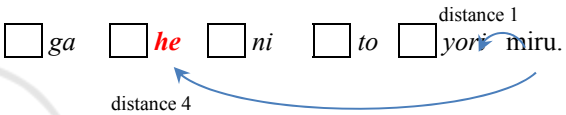
In Dataset 1, when generating an ungrammatical sentence, one case marker with distance 1 to 5 from a verb is replaced with an incorrect case marker which is not included in a correct set of case markers. An example of a grammatical and an ungrammatical sentence is as follows.

Dataset 1

- A grammatical sentence;

□ *ga* □ *wo* □ *ni* □ *to* □ *yori* miru.

- An ungrammatical sentence with distance 4;



The blanks □ are filled with an appropriate noun for each case markers. In the grammatical sentence of the fourth case "*wo*" from the verb "miru" in the grammatical sentence is replaced with an incorrect case marker "*he*" for the verb "miru".

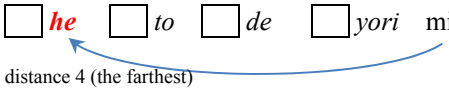
In Dataset 2, when generating an ungrammatical sentence, one case marker which is the farthest from a verb is replaced with an incorrect case marker which is not included in a correct set of case markers. An example of a grammatical and an ungrammatical sentence is as follows.

Dataset 2

- A grammatical sentence with 4 case markers;

□ *ga* □ *to* □ *de* □ *yori* miru.

- An ungrammatical sentence;



In the ungrammatical sentence, the farthest case marker from the verb "miru" is replaced with an incorrect case marker "*he*".

The difference between Dataset 1 and Dataset 2 is Dataset 1 is more informative rather than Dataset 2. Because there are some noun and case marker before the incorrect case marker in ungrammatical sentence

for Dataset 1, while there is no noun and case marker before the incorrect case marker for Dataset 2. When we evaluate the effect of distance, we can eliminate the effect of previous information of the incorrect case marker by applying Dataset 2.

For both Dataset 1 and Dataset 2, the verbs are selected from the highest frequency verbs recorded in the Kyoto University Case Frame Dictionary to generate sentences. And once the verb is selected, case markers are randomly selected from a correct set of case markers for the verb as illustrated in section 3. Also, when selecting appropriate nouns for the case markers, we refer to The Kyoto University Case Frame Dictionary where the noun and verb co-occurrence frequency is recorded in the dictionary. Then two highest frequency nouns are randomly selected for each case markers.

The number of a pair of a grammatical and an ungrammatical sentence are 320 in total for both Dataset 1 and Dataset 2.

Datasets replaced nouns with <unk>

In order to measure the effect of presence of nouns, we replace all the nouns accompanied by case markers with unknown words, <unk>. By using the example shown Dataset 1, we show an example blow. Here, the blank □ in Dataset 1 is replaced with <unk>. Note that the blank □ is filled with selected nouns in experiments.

Dataset 1 with nouns are replaced with <unk>

- A grammatical sentence;

<unk>*ga* <unk>*wo* <unk>*ni* <unk>*to* <unk>*yori* miru.

- An ungrammatical sentence with distance 4;

<unk>*ga* <unk>***he*** <unk>*ni* <unk>*to* <unk>*yori* miru.

When nouns are replaced with <unk>, the number of sentence pair is reduced to 10 – 240 for some distance. Because the combination of nouns, case marker, verb is lessened by replacing nouns with <unk>. In some datasets, by rearranging the case markers, the total number of sentences pair is 1,200.

4.2 Training Language Models

In this paper, we employed LSTM (Long short-term memory) for neural language model (Hochreiter et al., 1997) and n-gram model.

The Language models were trained by randomly selected 10,000 sentences from Japanese Wikipedia, and it contained about 220,000 words. Low-frequency words were replaced by the unknown word <unk>.

For the n-gram model, the kneser-ney method (Kneser and Ney, 1995) was applied for smoothing and the parameter n of n-gram model was set 2 and 5.

For the LSTM model, we employed the Keras for implementation. The middle layer consisted of two layers and each layer consisted of 650 nodes. The input was 30 string of words, and the words were embedded in 200 dimensions by using the Japanese Wikipedia entity vector by word2vec (Suzuki et al., 2016). RMSProp was applied for the gradient method, the learning rate was set to 0.001 and the dropout rate was 0.2. The number of epochs was set to 6 which is selected with the small datasets created for epoch evaluation, aside from the datasets for overall test evaluation reported in section 5.

4.3 Evaluation Language Models

A pair of a grammatical and ungrammatical sentence were applied to test the language models. Each word in a sentence was treated as an input to the models, then the generation probability of next word is calculated. After calculating the generation probability of the sentences, that is the joint probability of sequence of words, the generation probabilities for a pair of grammatical and ungrammatical sentences are compared. If the higher probability is assigned to the grammatical sentence than ungrammatical sentence, then the model prediction for the pair is correct. On the other hand, if the higher probability is assigned to the ungrammatical sentence than grammatical sentence, then the model prediction for the pair is wrong.

We report overall accuracies which count the number of correctly predicted pair of sentences divided by the total number of pair of sentences. We conducted this evaluation process for four datasets (Dataset 1, Dataset 2, Dataset 1 replaced the nouns with <unk> and Dataset 2 replaced the nouns with <unk>) and for LSTM and n-gram learning models.

5 RESULTS AND DISCUSSION

In this section, we evaluate neural language model (LSTM) with our evaluation datasets by comparing with n-gram models. We have four evaluation datasets, Dataset 1, Dataset 2, Dataset 1 replaced nouns with <unk> and Dataset 2 replaced nouns with <unk>. Figure 2 show the results.

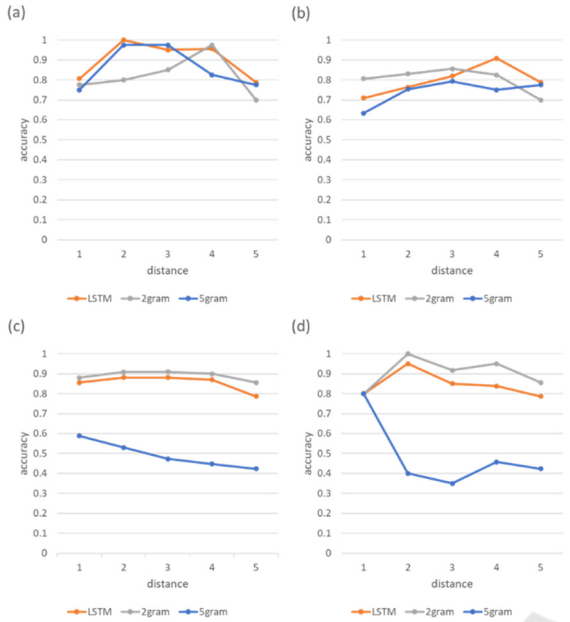


Figure 2: Evaluation performance for (a) Dataset 1, (b) Dataset 2, (c) Dataset 1 replaced nouns with <unk> and (d) Dataset 2 replaced nouns with <unk>. The distance 1 to 5 denotes the distance between the incorrect case marker and a verb.

Comparison of LSTM and 5-gram model

Figure 2 (a) and Figure 2 (b) corresponds to the results evaluated by datasets of Dataset 1 and Dataset 2. The Dataset 1 is fixed the number of case markers with 5 for each grammatical and ungrammatical sentence. For the Dataset 2, the number of case markers varies from 1 to 5. In Figure 2, the distance 1 to 5 refers to the distance of incorrect case marker from a verb in ungrammatical sentence.

As for the distance of incorrect case marker, distance 3-4 tend to exhibit better performance than other distance for both LSTM and n-grams models. This implies that to predict the incorrect case marker, it's should not be too close to the verb nor too far. For example, in distance 1, the predictions become worse because there is no information between the incorrect case and the verb. On the other hand, if it's too far, such as distance 5, the information can be deteriorated between the incorrect case and the verb.

The accuracies are the number of pairs which have correct answers divided by the total number of pairs. The Dataset 1 results exhibit better accuracies then Dataset 2. This indicates that even though Japanese sentence is head-final language, that is, the verb resides at the end of sentence and decides the case markers in the sentence, the previous information before the incorrect case marker, contributed to the prediction of case markers. Because the incorrect case is farthest position (resides as the first words in a sentence) in Dataset 2, so there is no information before for the incorrect case marker. On the other hand, in Dataset 1, there are several cases before the incorrect case marker and theses cases contributed for predicting the incorrect case even though proper case markers are determined by the verb grammatically.

As for the language models, LSTM tends to exhibit better or equal performance compared to n-gram models.

Figure 2 (c) and Figure 2 (d) are results for the Dataset 1 and Dataset 2 when the nouns are replaced with <unk>. When the nouns are all replaced with <unk>, then the 5-gram model largely drop their accuracies.
It indicates that the nouns are important keys to determine case. And when the nouns are not available, it's difficult for the n-grams to determine the case. On the other hand, LSTM performs better than 5-gram model even when the nouns are not available as clues to determine the case markers. LSTM can capture the case marker and verb relation, while n-gram models deteriorate their performance.

Problem of 2-gram model

In case of 2-gram, the performance seems to outperform LSTM and other n-gram models for all distances. However, 2-gram tend to return popular case markers, that is, case markers appeared in a corpus with high frequencies. For example, when we have a pair of sentences;

<unk> *ga* <unk> *to* miru. (grammatical sentence)
<unk> *he* <unk> *to* miru. (ungrammatical sentence)

The generation probabilities are compared with 2-gram of {(<unk>, *ga*) and (*ga*, <unk>)} and {(*ga*, <unk>) and (*he*, <unk>)} between sentences. Other 2-grams are same so that they don't contribute the difference on probabilities. Then, the 2-gram with *ga* case marker is higher probability than the 2-gram with *he* case markers, since *ga* case marker appears more often than *he* case marker in a corpus. In total, 2-gram is quite biased towards high frequency cases and this phenomenon contributed high accuracies in this experiment.

We examined above mentioned hypothesis that 2-gram tends to return high frequency case markers and obtains high accuracies. We observe the situation when low frequency case markers are assigned as grammatical sentences. If the 2-gram reject these grammatical sentences just because low frequency,

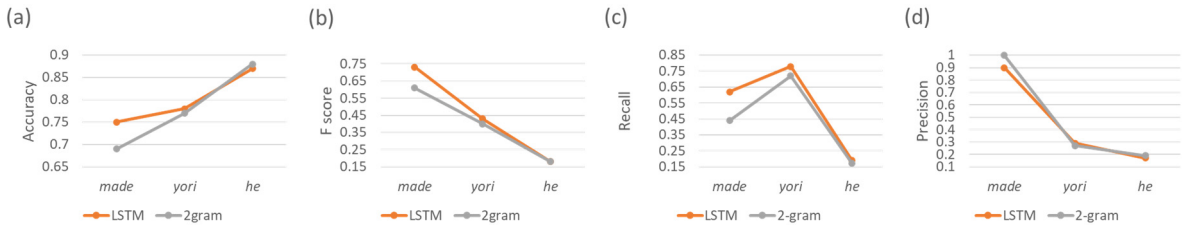


Figure 3: Evaluation of the low frequency case markers, "he", "made" and "yori". The figures show the performance measured by (a) accuracy, (b) F score, (c) Recall and (d) Precision, respectively.

then the performance should be worse for theses case markers.

For low frequency case markers, "*he*", "*made*", "*yori*" case markers are selected. We put all dataset and distance together because the size of dataset become too small to compare for these case markers. We compare the performance with F score, recall and precision for these case markers.

Figure 3 shows the result. We compared 2-gram with LSTM. While 2-gram performed better or equal to LSTM in accuracy, when compared in F score or recall for the low frequency case markers, the performance does not always outperform LSTM. This indicates that LSTM does not always return the popular cases and when looked at the low frequency case markers, LSTM tend to exhibit better in F score and recall.

In precision, "*made*" case marker exhibits highest accuracies for both LSTM and 2-gram. It is because the generation probability of the sentence which includes "*made*" becomes low, due to the low frequency of "*made*" case marker. This results in high precision and low recall as show in Figure 3. And this tendency is more obvious with 2-gram models than LSTM. So, 2-gram is more affected by the word frequency than LSTM. Overall, LSTM is more robust with low frequency words than n-gram models.

6 CONCLUSION

In order to evaluate the language model in Japanese, we proposed a language model evaluation method to assess Japanese grammar. We especially focused on Japanese case grammar of surface cases. We constructed evaluation datasets to assess case marker and verb matches in a sentence. To do so, we adopt target evaluation methods by generating grammatically correct sentence (grammatical sentence) and ungrammatical sentence as a pair of evaluation set.

Once the evaluation dataset is constructed, the learning models, neural language model (LSTM) and n-grams, are applied to the datasets and evaluate whether the language models could correctly estimate the relationship between the case marker and verb.

In the experiment, we also examined the effect of other information, such as nouns appear along the case markers. To remove the noun effect, we constructed the datasets by replacing all nouns with <unk>. When predicting the case marker and verb match, these nouns contributed for prediction performances. However, LSTM is less affected than n-grams by removing informative words. In this sense, LSTM is more robust than n-gram models.

LSTM is also less affected by frequent words than n-gram models. N-gram models, especially 2-gram, are almost only predicts the frequent words (case markers) as the correct ones. It can achieve high accuracy, however results in low F score or recall performance. When precise analysis is required, such as, good performance for low frequent words (or case markers), LSTM is better learning model.

REFERENCES

Fillmore, C.J., 1971. Some Problems for Case Grammar. Report on the 22nd Round Table Meeting on Linguistics and Language Studies, ed. by Richard J. O'Brien, S.J., 35-56. Georgetown: Georgetown University Press.

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M., 2018. Colorless Green Recurrent Networks Dream Hierarchically. In Proceedings of the 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp.1195–1205.

Hasegawa, S., Kikuchi, Y., Takamura, H., Okumura, M., 2017. Japanese sentence compression with a large training dataset. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers. pp. 281-286.

Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural Computation, 9(8):1735–1780.

Imamura, K., Fujita, A., Sumita, E., 2018. Enhancement of encoder and attention using target monolingual corpora

in neural machine translation. In Proceedings of the 2nd Workshop on Neural Machine Translation and Generation. pp. 55-63.

Kawahara, D. and Kurohashi, S., 2001. Japanese Case Frame Construction by Coupling the Verb and its Closest Case Component. In Proceedings of the Human Language Technology Conference, pp.204-210.

Kawahara, D. and Kurohashi, S., 2002. Fertilization of case frame dictionary for robust Japanese case analysis. In Proceedings of the 19th International Conference on Computational Linguistics, pages 425–431.

Kawahara, D. and Kurohashi, S., 2006a. A Fully-Lexicalized Probabilistic Model for Japanese Syntactic and Case Structure Analysis. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL2006), pp.176-183.

Kawahara, D. and Kurohashi, S., 2006b. Case Frame Compilation from the Web using High-Performance Computing, In Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006).

Kim, Y., M. Rush, A., Yu, L., Kuncoro, A., Dyer, C., and Melis, G., 2019. Unsupervised recurrent neural network grammars. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, pages 1105–1117, 2019.

Kneser, R., Ney, H. 1995. Improved backing-off for m-gram language modeling. In 1995 International Conference on Acoustics, Speech, and Signal Processing, Vol. 1, pp. 181-184.

Linzen, T., Dupoux, E., and Goldberg, Y., 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. In Transactions of the Association for Computational Linguistics (TACL), Vol. 4, pp. 521–535.

Marvin, R., Linzen, T., 2018. Targeted Syntactic Evaluation of Language Models. In Empirical Methods in Natural Language Processing (EMNLP), pp. 1192–1202 Brussels, Belgium.

Miyazaki, T., Shimizu, N., 2016. Cross-lingual image caption generation. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Volume 1: Long Papers. pp. 1780-1790.

National Institute for Japanese Language, 1997. Correspondence between surface and deep case in Japanese", National Institute for Japanese Language Report; 113, National Institute for Japanese Language Academic Information Repository. Available from: http://doi.org/10.15084/00001282 (in Japanese)

Nochi, H., Takamura, D., 2019. Improving grammar ability of language model by distinguishing from explicit non-sentence. In Proceedeings of the 25th The Association for Natural Language Processing (NLP2019), no.B5-2, pp.962-965. (in Japanese)

Papineni, K., Roukos, S., Ward, T., Zhu, W. J., 2002. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311-318.

Sennrich, R., 2017. How grammatical is character-level neural machine translation? Assessing MT quality with contrastive translation pairs. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 376–382. Association for Computational Linguistics.

Suzuki, M., Matsuda, K., Sekine, S.,Okazaki, N., Inui, Kentaro., 2016. Multiple assignment of extended property expression labels to Wikipedia articles" In Proceedings of the 22th The Association for Natural Language Processing (NLP2016), 4p.(in Japanese)

Warstadt, A., Singh, A. and Bowman, S. R., 2019. Neural Network Acceptability Judgments, Transactions of the Association for Computational Linguistics, Volume 7, p.625-641.

Zweig, G., Burges, C. J., 2011. The microsoft research sentence completion challenge. Microsoft Research, Redmond, WA, USA, Tech. Rep. MSR-TR-2011-129.