Article

Developing Language-Specific Models Using a Neural Architecture Search

YongSuk Yoo [1] and **Kang-moon Park [2,*]**

1   Department of English Literature, College of Humanities, Jeonbuk National University, Jeonju-si 54896, Korea; ysyoo@jbnu.ac.kr
2   Department of Elctronic Engineering, Korea National University of Transportation, Chungju 27469, Chungcheongbuk-do, Korea
*   Correspondence: kmpark@ut.ac.kr; Tel.: +82-42-841-5367

Featured Application: Neural Architecture Search (NAS) on linguistic tasks.

Abstract: This paper applies the neural architecture search (NAS) method to Korean and English grammaticality judgment tasks. Based on the previous research, which only discusses the application of NAS on a Korean dataset, we extend the method to English grammatical tasks and compare the resulting two architectures from Korean and English. Since complex syntactic operations exist beneath the word order that is computed, the two different resulting architectures out of the automated NAS language modeling provide an interesting testbed for future research. To the extent of our knowledge, the methodology adopted here has not been tested in the literature. Crucially, the resulting structure of the NAS application shows an unexpected design for human experts. Furthermore, NAS has generated different models for Korean and English, which have different syntactic operations.

Keywords: deep learning; neural architecture search; word ordering; Korean syntax

1. Introduction

We show an interesting result of the application of a modified neural architecture search (NAS) in [1] to linguistic tasks (grammaticality judgment) for Korean and English syntactic phenomena. Based on the previous research on this subject in [2], we show that the extension of the NAS method to English grammatical tasks provides a different architecture from the one generated for the Korean dataset. This is rather unexpected given the similarity of the input data. The major contribution of this paper is to show that the previous application of NAS to linguistically complex datasets of Korean [2] can be extended to the linguistic phenomena of English. Notably, the different resulting architecture in these two experiments clearly indicates that the NAS method is sensitive to the different word order that contains multiple syntactic operations. The scientific purpose of this paper is to develop language models using NAS.

Deep learning has been applied successfully in various fields due to its powerful performance on difficult problems and pattern findings [3,4], such as image recognition [5,6] and natural language processing (NLP) [7,8]. Importantly, the application of deep learning methods to the field of psycholinguistics has been successful [9]. As noted in the literature, the understanding of psycholinguistics in terms of the deep learning method may show how languages can be processed computationally. NAS aims to automate the architecture engineering, which can be applied to various fields [10–12]. Although all the designs can be created manually, researchers have suggested an automated design process that can be efficient in various applications. NAS methods have shown successful results in various fields of studies, such as including image classification [13,14], object detection [15], or semantic segmentation [16]. The upshot of this is to reduce errors [17] that could

happen when we design architecture manually and make the automation to search for the best-performing learning algorithm [18].

However, we take a different perspective on the application of NAS to the linguistic phenomenon. The main goal of this paper is to explore the resulting architecture out of NAS application in various linguistic data that contain different syntactic operations. The previous research of NAS application to linguistic data focused on the improvement of accuracy compared to existing language models [19]. However, as they noted in the article, the research is somewhat limited as NAS does not provide a better language model. In this experiment, we will compare the resulting architecture of the Korean grammaticality judgment dataset to the architecture of the English grammaticality judgement dataset. Given that Korean and English have very different linguistic properties, we predict that NAS will generate different architecture that is fitting to each dataset, presented out.

In our previous research [2] on Korean grammatical tasks, we applied the NAS method to word order patterns found in Korean. Word order patterns of Korean involve operations called ellipsis and scrambling, which add complexity to the dataset [20]. A lot of deep learning studies have proven that the word order tasks can be performed without any explicit syntactic information [21]. However, their research is somewhat limited in that they focus on the accuracy of a specific model. Given that the syntactic information may not be necessary for the deep learning task, the improved accuracy of a specific model does not guarantee that the language model is better.

The application of NAS method to Korean grammaticality patterns involving scrambling (1) and ellipsis (2) provides an architecture which was discussed in our previous paper [2], which we will discuss in detail in Section 2.

(1)

a. John-i	Mary-lul	coahanta
John-subject	Mary-object	like
'John likes Mary'		
b. Mary-lul	Johin-i	coahanta
Mary-object	John-subject	like
'John likes Mary'		

(2)

a. (John-i)	Mary-lul	coahanta
b. John-i	(Mary-lul)	coahanta
c. (John-i)	(Mary-lul)	coahanta

As we noted in the paper, the two linguistic phenomena in Korean add complexity to the dataset: (i) scrambling, which allows different ordering patterns of inputs (1) [22], and (ii) argument ellipsis which involves invisible element in their sentence (2). Note here that these two operations are not available for English. NAS has provided a model that successfully learns the grammaticality of the Korean dataset.

This paper extends the NAS application to English data patterns. We applied the NAS method with the English dataset that has different grammatical properties from Korean. For example, English allows so-called verb fronting, which puts the verb in front of other items, as shown in (3) [23].

(3)   pass one now he has.

In this paper, we report that the application of NAS into Korean and English grammaticality tasks yields two different resulting architectures. To our knowledge, the finding sheds new light on the research of language modeling since the automation of architecture is sensitive to the grammatical information underlying the word order of languages.

The organization of this paper is as follows: in Section 2 we briefly review the results of the previous research [1]. In Section 3 we show the methodology. We show the result of the experiment in Section 4. Finally, Section 5 concludes this paper.

2. Review of the Previous Research on Korean Grammaticality Judgement Tasks

2.1. Korean Grammaticality Judgement Task

This section discusses the dataset that was researched in our previous article for Korean grammaticality tasks. The input data consist of four words with seven syntactic categories ($7 \times 7 \times 7 \times 7$). The input data are one-dimensional; however, due to the property of language, the information beneath the word order is complex. This is because of Linearization, which is a process of generating grammatical word orders for a given set of words. Even though the input words are limited, the language specific processes add complexity to the dataset. In this experiment, we have considered syntactic processes called ellipsis and scrambling. Ellipsis refers to the syntactic phenomenon in Korean which licenses a null element to be grammatical, as shown in (1a) (The elided elements are presented under the strikethrough). The availability of ellipsis of an element in Korean thus shows that the four-word sentences can carry a virtually unlimited number of hidden words. Note here that the input for the NAS method is represented as syntactic categories, instead of an individual word. Scrambling is a process which allows different ordering of elements. The exact mechanism behind scrambling is very intricate, in that only a limited number of the combinations are available in a language. For example, for the sentence in (1a), two other combinations are grammatical. (1b) represents one case of scrambling.

(1)

a. Jane-i	yepputako	John-i	cipeysey	malhaytta.
Jane	pretty	John	home	said
'At home, John said that Jane is pretty.'				
b. John-i Jane-i yepputako cipeysey malhayta.				

Among the 2401 combinations of seven syntactic categories ($7 \times 7 \times 7 \times 7$), 113 sentences turn out to be grammatical. The data are checked by two trained linguists who are Korean native speakers. The distribution of the data is given in Figure 1. Each axis, including the color bar, represents the word slot. The numbers represents the syntactic categories, as illustrated in Table 1. The circle in the distribution means that the syntactic combination from the four axes are grammatical. Table 2 shows examples of sentences.
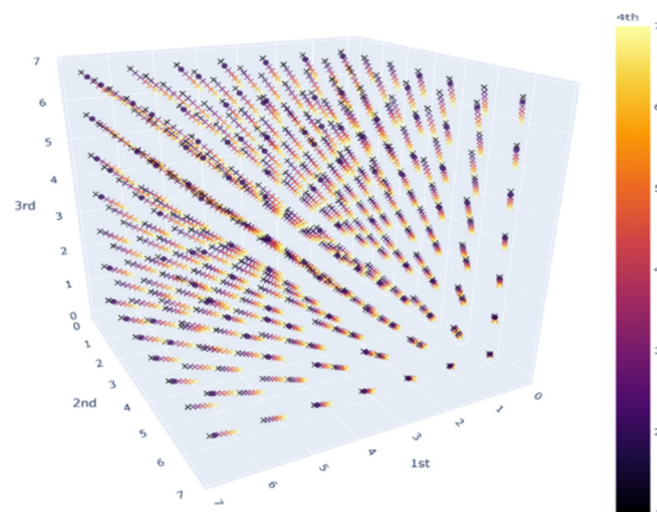

Figure 1. Distribution of Korean dataset.

Table 1. Syntactic categories.

Parameter	Value
NP	1
VP	2
AP	3
CP	4
ADVP	5
AUX	6
PP	7

Table 2. Examples of sentences.

	1st	2nd	3rd	4th	Correcteness
Sentence 1	NP(1)	VP(2)	CP(4)	AUX(6)	X
Sentence 2	NP(1)	NP(1)	NP(1)	NP(1)	X
Sentence 3	NP(1)	NP(1)	NP(1)	VP(2)	O

2.2. Neural Architecture Search (NAS)

This section only provides a brief introduction of the neural architecture search (NAS). We refer readers to our previous paper [1] for a more detailed discussion. For ease of exposition, we will focus on the three main stages of NAS: search space, search strategy, and performance estimation strategy [17,24]. The search space defines an architecture. We need to input the prior knowledge into the system to improve efficiency. The search strategy determines the overall shape of the architecture. NAS will generate multiple candidates here that are suitable for the given dataset. In the performance estimation strategy, the best performing architecture will be decided and be provided as the resulting architecture.

While there are different types of NAS methods available, we apply the Evolutionary Algorithm (EA) in this experiment [25,26]. The upshot of this method is that it does not distinguish a minimum architecture from a middle architecture. Since we are in search of a language model that has not been tested before, this insensitivity to the input structure provides advantages for the current purpose of this paper. In particular, we are using the so-called variable chromosome genetic algorithm (VCGA) proposed in [1], which is one of the modified versions of EA. This method eliminates the necessity of minimum architecture since its genetic operation uses destructive methods as well as constructive methods. We refer readers to [1] for the in-depth discussion of this method compared to other NAS methodologies.

2.3. NAS Method to Korean Grammaticality Judgement

In our previous research [2], the NAS method successfully provided architecture that captured the word order patterns in Korean, as shown in Figure 2. Figure 2 shows the final architecture of the Korean grammaticality task. It has the same number of layers with initial architecture. However, it has multiple links between the hidden layer and the output layer.
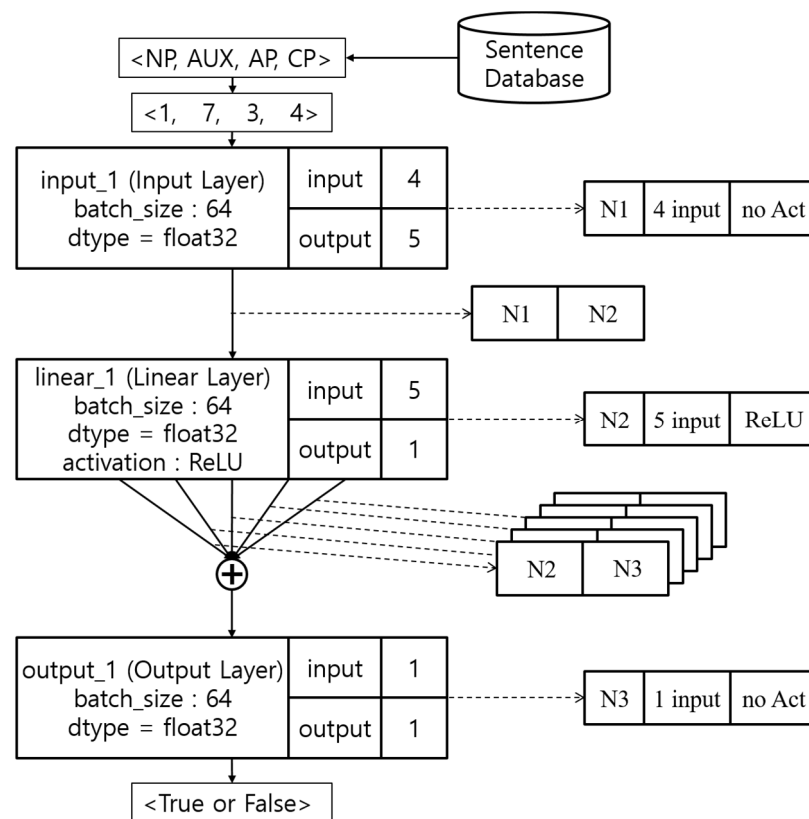
Figure 2. Final structure of Korean grammaticality task.

The resulting topology involves the five outputs of the hidden layer that are added into the one input of the output layer. This is unexpected as in general, one-to-one correlation would be adopted. The accuracy rate indicates that the resulting architecture is efficient enough for the given dataset. The application of NAS method seems to be successful for the given task.

In the previous research [2], we argued that this topology is specific to the Korean grammaticality patterns; however, we could not prove this due to the absence of a minimal pair. This is the main reason why we extend the experiment to English data patterns, which have different sets of syntactic operations. If the experiment provides different topologies, we can argue that the NAS method is indeed sensitive to the syntactic operations that underlie in the word order patterns. The experiment of this paper shows that the NAS method is indeed sensitive to the grammatical differences in the word order patterns. We will provide comparisons of the datasets and resulting architectures in the next section.

3. NAS Method to English Grammaticality Judgement

We replicated the experiment of the previous research [2] to the English dataset. As mentioned before, the particular method is called VCGA. It is shown in Figure 3. Since the main goal of this paper is to compare the resulting architecture between Korean and English, a shortened introduction of the method is provided. We refer readers to [2] for a more detailed discussion of the method. As mentioned before, the upshot of VCGA is to involve destructive searching [1]. The operation by the chromosome non-disjunction allows multiple generations of ANN architectures. Due to its property, the final result is identical regardless of the initial status of the input structure. This method consists of three phases; in the first phase, NN generator design NN based on chromosomes through model checker and link checker; in phase 2 generated NNs are trained and validated with the inputted dataset; in phase 3 genetic operators select individuals to survive and make offspring based on survived individuals as parents. More details of this method are expressed in previous papers [1,2]. The group of generators including a genetic algorithm

generator and neural networks generators ensures the following properties: (i) a cross-over operation blending information of a parent for various offspring (ii) a mutation operation; and (iii) a non-disjunction operation [1] making the distinction between the two offspring by less or more information. These operators change hyperparameters such as composition of layers, linkage, the number of nodes, activation function, etc. [1].
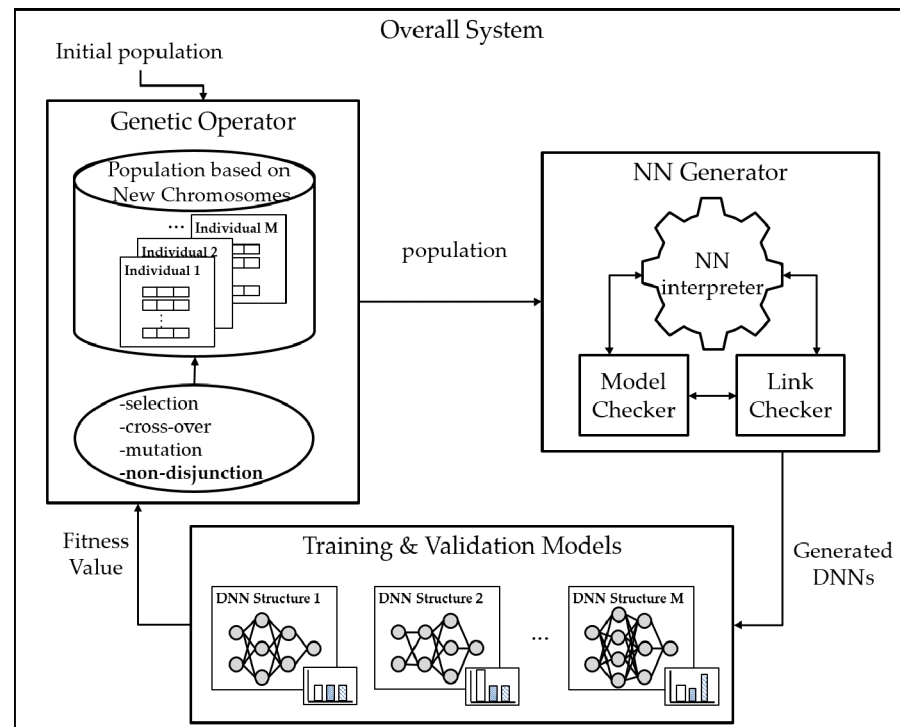

Figure 3. System of modified NAS method.

In order to apply this method to the English case, we form a dataset which consists of English grammatical tasks with four words. The data are expressed as a combination of some digits, according to their grammatical categories (1: Noun; 2: Verb; 3: Adjectives; etc.). For example, the sentence 'John likes beautiful Mary' is expressed as '1, 2, 3, 1' As such, the grammatical and non-grammatical sentences of all cases are expressed in digits.

We labeled this dataset based on whether they are grammatical or non-grammatical. We used this dataset as inputs of the NAS algorithm, similar to previous research [2]. We intentionally created a similar dataset in order to compare the resulting structure between Korean and English. The data consist of seven syntactic categories for four-word level sentences. We have obtained 2401 combinations of syntactic categories, and we consulted the grammaticality of the sentences with three linguistically trained native speakers of English. We plan to share the database upon the publication of this paper. The distribution of the grammaticality of the English dataset is given in Figure 4, and the data are treated as training data for the NAS, and it is tuned to generate a neural architecture for grammaticality judgment of the English dataset.
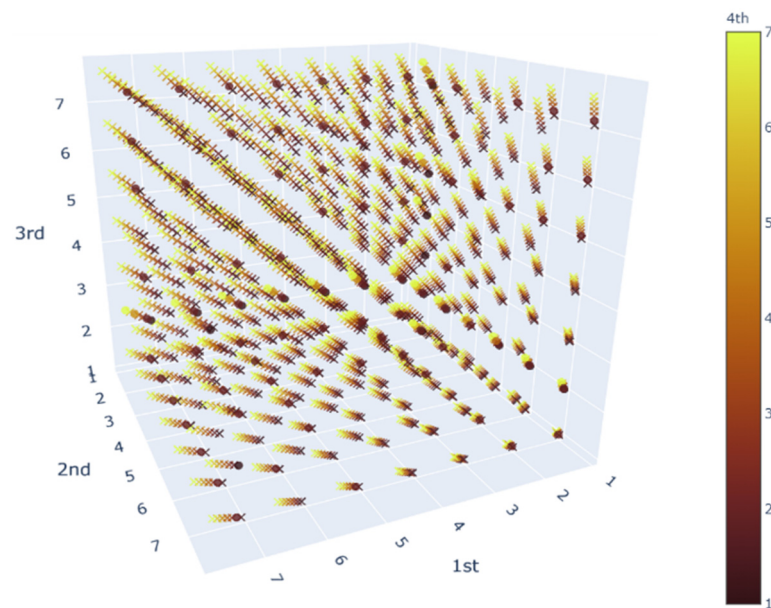
Figure 4. Distribution of English dataset.

4. Experiment

4.1. English Grammaticality Judgement

The basic structure of the experiment is identical to the previous experiment on Korean. The main goal behind this design is to compare the resulting structures of NAS application on Korean and English grammaticality judgement tasks. In this experiment, we have created four-word level sentences with seven syntactic categories: noun, verb, preposition, adjective, adverbs, complementizer, and auxiliary phrases.

The grammar of English is radically different from that of Korean [27]. However, the different grammar is only expressed on the linear order of the word inputs. Thus on the input level it seems very similar as the only difference here is the number of the correct sentences by different combinations of syntactic categories.

In detail, the verbs in Korean must come at the end of the sentence, whereas English allows the verb to appear with a major degree of placement [28]. The dataset in question consists of 2401 combinations, where 136 sentences are grammatical. The dataset was consulted with two linguists who are native English speakers. In comparison to the Korean dataset, the overlapping cases were 53. The first slot of four words is expressed in the X-axis, and Y and Z, respectively, represent the second and the third slot. The fourth word slot is represented by the color spectrum. The O/X represents grammaticality. Figure 4 shows the distribution of the English grammaticality dataset.

We used a fitness function to determine the next generation on the genetic algorithm. Fitness function (Equation (1)) is defined as follows:

$$fitness = (1 - \text{loss}) - R * \frac{num_{layer}}{num_{avg}} * (2 * (1 - loss) - 1) \tag{1}$$

4.2. Experiment Setups

The experiment setup for English grammaticality tasks is identical to the previous experiment for the Korean dataset. We carefully controlled the system in that the resulting architecture of NAS application is able to be compared. The diagram for the initial network is given in Figure 5: an input layer, a hidden layer, and an output layer. There are five nodes in the hidden layer. Rectified Linear Unit (ReLU) functions as an activation function. The loss of the initial neural architecture model is about 0.002338. Parameters of these experiments are shown in Table 3.
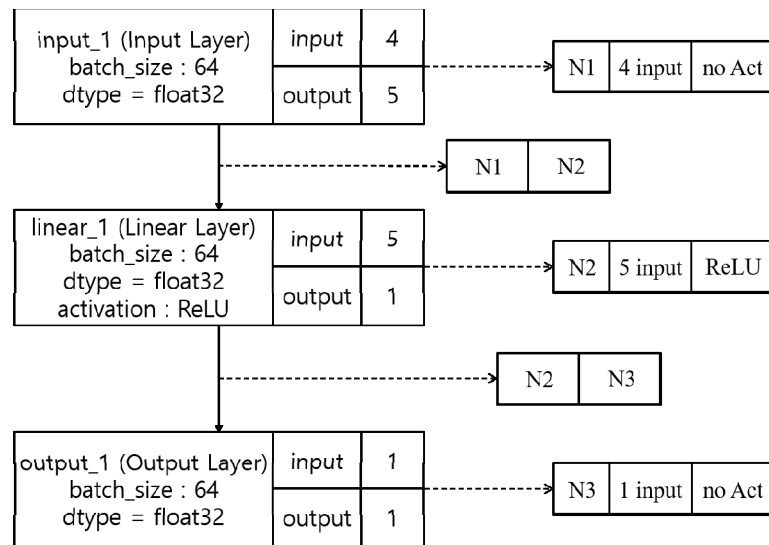
Figure 5. Initial architecture.

Table 3. Parameters of Experiment.

Parameter	Value
Population	50
Generations	30
Mutation rate	0.05
Crossover rate	0.05
Non-disjunction rate	0.1
Learning rate	0.01
Loss function	MSE Loss

All neural networks in our search space are composed of linear layers with identical structure but with different weights. The proposed NAS algorithm searches neural architectures using VCGA [1] which optimizes overall structure, including composition of layers, connections between layers, the number of nodes and activation function, using input neural networks. In order to optimize the initial neural network, we use the number of chromosomes and loss value of generated neural networks as the fitness value and generated neural networks use Korean and English datasets.

4.3. Experiment Results

The experiment result is interesting. The resulting architecture for the English dataset is radically different from the one for the Korean dataset, despite their distributional similarity; Korean has 113 grammatical sentences, and English has 136, as shown in Table 3. The evolution process of the experiment is presented in Figure 6; it starts with five chromosomes within three layers, and it evolves into three chromosomes.
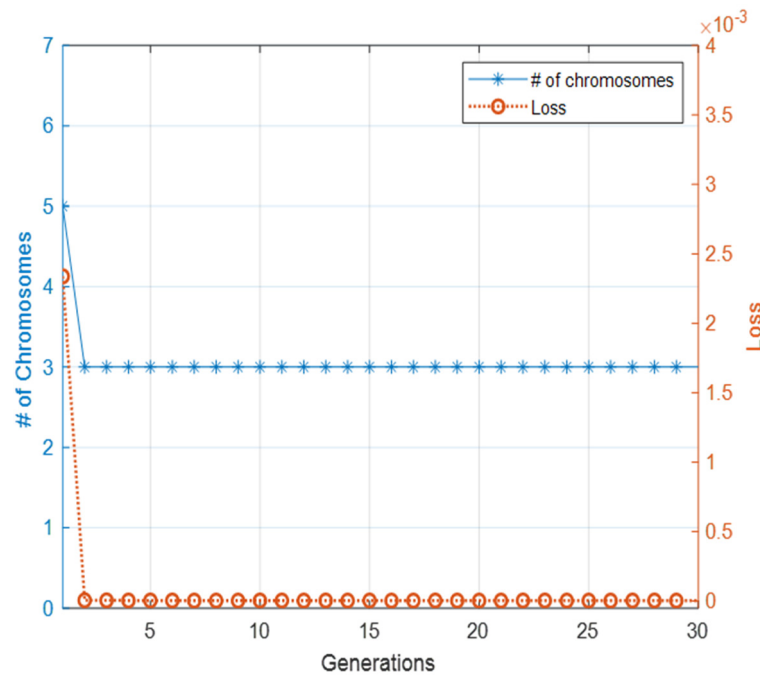
Figure 6. Evolution process of English grammaticality task.

The loss is reduced from 0.002338 to 0.000004 during this process. The final architecture does not have any hidden layer between the input and output layers. The resulting topology given in Figure 7 is interesting. This network calculated four-word ordering in an English grammaticality task without a hidden layer. This was very different from the results of the Korean grammaticality task.
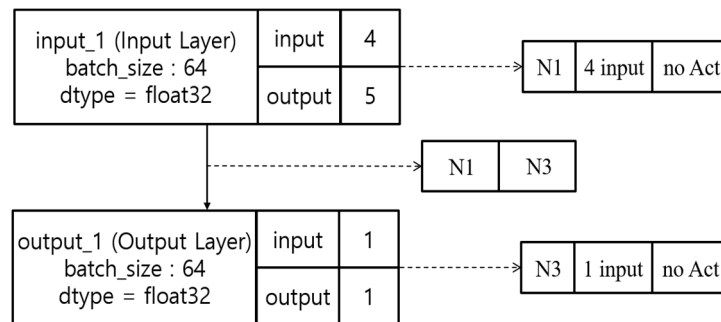

Figure 7. Resulting topology of English Grammaticality Task.

The resulting topology is surprising, given that the dataset for Korean and English is almost identical. The distribution of these two types of data seems to be similar. There are 53 of 2401 matching cases for 113 and 136 grammatical sentences in the respective laguages.

Figure 8 compares generated ANN architecture of the Korean grammaticality task and the English grammaticality task. Figure 8a presents the generated architecture of the Korean grammaticality task. It has one hidden layer with five nodes and four additional links between the hidden layer with ReLU as an activation function and the output layer, instead of leaky ReLU, because of calculation speed. Each layer uses float 32 data type. Figure 8b presents the generated architecture of the English grammaticality task. It has an input layer with four nodes without a hidden layer.
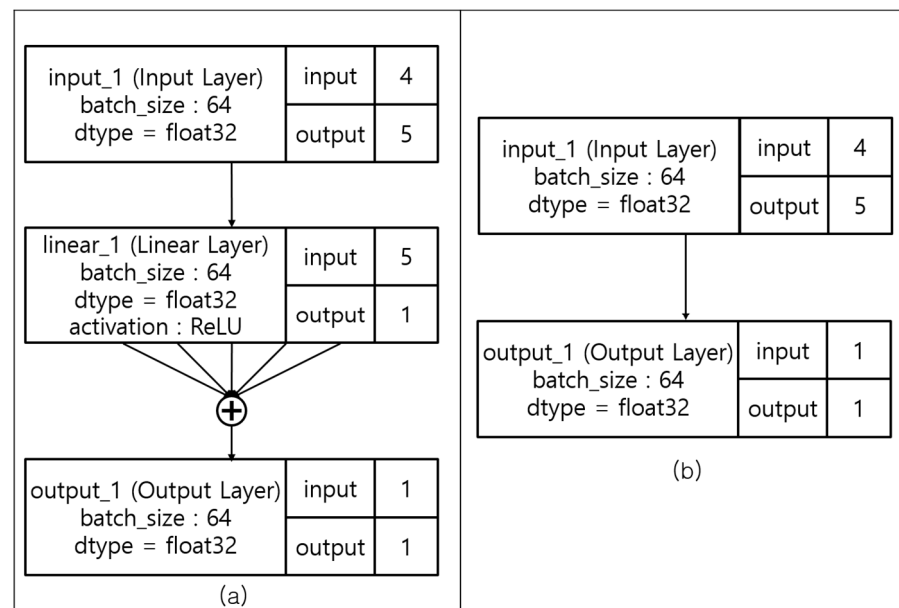
Figure 8. Comparison between Korean and English grammaticality task topology. (a) the generated architecture of the Korean grammaticality task; (b) the generated architecture of the English grammaticality task.

Table 4 shows a summary of Korean and English grammaticality judgment tasks. English patterns have more grammatical combinations. Nevertheless, the complexity of the resulting neural architecture is less complicated compared to Korean.

Table 4. Comparison between Korean and English Tasks.

	Korean	English
Number of correct sentences	113	136
Number of layers	5	3
Average loss	0.000096	0.000003

We varied the number of layers to verify the resulting structure. We also conducted an experiment with a randomized initial population, which also provides the identical results. Hyperparameters are controlled throughout the experiment. The batch size of this experiment is 64 with 20 epochs, and learning rate is 0.0002. GTX 1660ti is used for this experiment. Each layer is fully connected with five nodes. The activation function used ReLU.

We argue that the two different resulting structures between Korean and English captures the linguistic differences that are underlying beneath the word order patterns. For example, Korean word-order patterns contain the argument ellipsis operation (2) which means that the four-word sentences can involve more than four words in terms of their syntactic structure. Crucially, English does not have the counterpart of this operation. However, we do not insist that the single different operation in the syntactic operation would be directly connected to the number of layers, as English also has exclusive syntactic operation that is not available in Korean. However, the current experiment clearly indicates that the different syntactic operations can be detected by the NAS method.

5. Discussion and Conclusions

The results of this experiment show that NAS application in linguistic tasks is successful in two respects: (i) the NAS application easily finds the efficient language model for the given task; (ii) the NAS application is sensitive to the grammatical differences existing in the word order patterns. In other words, the searching process of NAS can provide

interesting aspects of language modeling in that it provides different designs for different languages. Crucially, this work may also contribute to the field of computational psycholinguistics; as a result, it could be related to the black box problem of language models. The different resulting architecture indicates that the NAS method indeed creates a design that the human expert would not propose. In further research, we will enlarge the database in Korean and English, in addition to expanding the experiment to other languages. We expect that linguistically similar languages will have similar resulting architecture.

The limitation of this research needs to be clearly stated. The first issue is the size of the dataset. Since the entire database has to be checked manually by individual linguists, it requires more time to expand the data. We predict NAS is sensitive to the syntactic operations, thus the size would not affect the result, yet we still need to expand the dataset to confirm the resulting architecture. The second issue is to develop a methodology to compare resulting structures, and to understand the implication of it. We plan to add a third language to this experiment to investigate this issue.

Particularly, Japanese—which also has ellipsis and scrambling—is an interesting language to compare with Korean. We expect the NAS to generate a similar topology as a result. In further research, we will extend the experiment to Japanese by forming a relevant dataset.

References

1. Park, K.; Shin, D.; Chi, S. Variable Chromosome Genetic Algorithm for Structure Learning in Neural Networks to Imitate Human Brain. *Appl. Sci.* **2019**, *9*, 3176. [CrossRef]
2. Park, K.; Shin, D.; Yoo, Y. Evolutionary Neural Architecture Search (NAS) Using Chromosome Non-Disjunction for Korean Grammaticality Tasks. *Appl. Sci.* **2020**, *10*, 3457. [CrossRef]
3. Wang, T.; Wen, C.-K.; Jin, S.; Li, G.Y. Deep learning-based CSI feedback approach for time-varying massive MIMO channels. *IEEE Wirel. Commun. Lett.* **2018**, *8*, 416–419. [CrossRef]
4. Hohman, F.; Kahng, M.; Pienta, R.; Chau, D.H. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Trans. Vis. Comput. Graph.* **2018**, *25*, 2674–2693. [CrossRef]
5. Li, A.A.; Trappey, A.J.; Trappey, C.V.; Fan, C.Y. E-discover State-of-the-art Research Trends of Deep Learning for Computer Vision. In Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Bari, Italy, 6–9 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1360–1365.
6. Han, X.; Laga, H.; Bennamoun, M. Image-based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1578–1604. [CrossRef]
7. Lopez, M.M.; Kalita, J. Deep Learning applied to NLP. *arXiv* **2017**, arXiv:1703.03091.
8. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
9. Linzen, T.; Dupoux, E.; Goldberg, Y. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 521–535. [CrossRef]
10. Hatcher, W.G.; Yu, W. A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access* **2018**, *6*, 24411–24432. [CrossRef]
11. Simhambhatla, R.; Okiah, K.; Kuchkula, S.; Slater, R. Self-Driving Cars: Evaluation of Deep Learning Techniques for Object Detection in Different Driving Conditions. *SMU Data Sci. Rev.* **2019**, *2*, 23.
12. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]

13. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8697–8710.

14. Real, E.; Aggarwal, A.; Huang, Y.; Le, Q.V. Regularized evolution for image classifier architecture search. In Proceedings of the Aaai Conference on Artificial Intelligence, Honolulu, HI, USA, 2–9 February 2019; Volume 33, pp. 4780–4789.

15. Zoph, B.; Cubuk, E.D.; Ghiasi, G.; Lin, T.-Y.; Shlens, J.; Le, Q.V. Learning data augmentation strategies for object detection. *arXiv* **2019**, arXiv:1906.11172.

16. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.

17. Elsken, T.; Metzen, J.H.; Hutter, F. Neural architecture search: A survey. *arXiv* **2018**, arXiv:1808.05377.

18. Wong, C.; Houlsby, N.; Lu, Y.; Gesmundo, A. Transfer learning with neural automl. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 8356–8365.

19. Jiang, Y.; Hu, C.; Xiao, T.; Zhang, C.; Zhu, J. Improved differentiable architecture search for language modeling and named entity recognition. 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), Hongkong, China, 3–7 December 2019; pp. 3576–3581.

20. Saito, M. Optional A-scrambling. *Jpn. Korean Linguist.* **2009**, *16*, 44–63.

21. Schmaltz, A.; Kim, Y.; Rush, A.M.; Shieber, S.M. Adapting sequence models for sentence correction. *arXiv* **2017**, arXiv:1707.09067.

22. Saito, M. Some asymmetries in Japanese and their theoretical implications. PhD Thesis, MIT, Cambridge, MA, USA, 1985.

23. Chomsky, N. *The Minimalist Program*; Massachusetts Institute of Technology: Cambridge, MA, USA, 1995.

24. Weng, Y.; Zhou, T.; Li, Y.; Qiu, X. NAS-Unet: Neural architecture search for medical image segmentation. *IEEE Access* **2019**, *7*, 44247–44257. [CrossRef]

25. Back, T. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*; Oxford University Press: Oxford, UK, 1996.

26. Schmitt, L.M. Theory of genetic algorithms. *Theor. Comput. Sci.* **2001**, *259*, 1–61. [CrossRef]

27. Ahn, S.-H. Korean quantification and Universal Grammar. Ph.D. Thesis, University of Connecticut, Mansfield, CT, USA, 1991.

28. Nam, K.S.; Ko, Y.G. *The Standard Korean Grammar*; Top Publishing Company: Seoul, Korea, 1993.