Article

Global and Local Information Adjustment for Semantic Similarity Evaluation

Tak-Sung Heo ^{1,2}, Jong-Dae Kim ^{1,2}, Chan-Young Park ^{1,2} and Yu-Seop Kim ^{1,2,*}

¹ Department of Convergence Software, Hallym University, Chuncheon-si 24252, Gangwon-do, Korea; gjxkrtd221@gmail.com (T.-S.H.); kimjd@hallym.ac.kr (J.-D.K.); cypark@hallym.ac.kr (C.-Y.P.)

² BIT Research Center, Hallym University, Chuncheon-si 24252, Gangwon-do, Korea

* Correspondence: yskim01@hallym.ac.kr; Tel.: +82-10-2901-7043

Abstract: Semantic similarity evaluation is used in various fields such as question-and-answering and plagiarism testing, and many studies have been conducted into this problem. In previous studies using neural networks to evaluate semantic similarity, similarity has been measured using global information of sentence pairs. However, since sentences do not only have one meaning but a variety of meanings, using only global information can have a negative effect on performance improvement. Therefore, in this study, we propose a model that uses global information and local information simultaneously to evaluate the semantic similarity of sentence pairs. The proposed model can adjust whether to focus more on global information or local information through a weight parameter. As a result of the experiment, the proposed model can show that the accuracy is higher than existing models that use only global information.

Keywords: semantic similarity; Siamese network; bidirectional long short-term memory; self-attention; capsule network; Manhattan distance

Citation: Heo, T.-S.; Kim, J.-D.; Park, C.-Y.; Kim, Y.-S. Global and Local Information Adjustment for Semantic Similarity Evaluation. *Appl. Sci.* **2021**, *11*, 2161. <https://doi.org/10.3390/app11052161>

Academic Editor: Wen-Hsiang Hsieh

Received: 8 December 2020

Accepted: 25 February 2021

Published: 1 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Semantic similarity evaluation is used in various fields such as machine translation, information retrieval, question-and-answering, and plagiarism detection [1–4]. Semantic similarity is measured for two texts, regardless of the length, the location of the corresponding words, and their contexts. These semantic similarity evaluations cost a lot of time and money in order for a person to judge directly. To solve this problem, past studies have used bilingual evaluation understudy (BLEU) [5] or metric for evaluation of translation with explicit ordering (METEOR) [6]. However, these are both vocabulary-based similarity evaluation methods, and it is difficult to grasp similar expressions, and not just the same ones. Recent studies [2–4] have shown good performance using artificial neural networks, such as convolutional neural networks (CNN), long short-term memory (LSTM) and gated recurrent unit (GRU), to evaluate the semantic similarity of sentence pairs.

These methods [2–4] have used the output of the last hidden state, which represents a whole sentence, to evaluate similarity. However, if sentence similarity is judged using only information representing the entire sentence, there is a limitation, in that that it is difficult to properly reflect the effect of similarity of local meaning [7,8]. If a sentence is composed of multiple sentences, it is very important to estimate the meaning similarity between individual sentence pairs. Let us take an example below.

Sentence 1: I'm a 19-year-old. How can I improve my skills or what should I do to become an entrepreneur in the next few years?

Sentence 2: I am a 19 years old guy How can I become a billionaire in the next 10 years?

The two sentences above have similar expressions overall, but each second sentence has a distinctly different meaning. Therefore, in this study, we propose a model that uses not only global features, i.e., entire information on sentences, but also local features, i.e., local information on sentences.

Recently, the capsule network [9] has been proposed that showed good performance using local features in the image classification task. This approach vectorizes the properties of the entity, considering spatial relationships. A capsule network has also shown good performance in text classification tasks [10–12]. A capsule network finally considers the spatial relationship of local features, and more elaborated learning is achieved through dynamic routing that sends input values to the appropriate upper-level capsules. However, in the case of the semantic similarity task, even if the order of words in the two texts were different, it cannot be asserted that the meaning of the two texts is different. Therefore, in this study, the use of dynamic routing is avoided, and the individual features that constitute the capsule are used as local features.

We propose a model that uses global and local features together for semantic similarity evaluation of sentence pairs. The two sentences are inputted into bidirectional LSTM (Bi-LSTM), which has a forward-backward LSTM structure [13], and the last hidden states extracted through this are used as global features. The sequence of hidden states extracted from Bi-LSTM is entered into self-attention to express the sentence's context information [8,14–16]. The features from Bi-LSTM are entered into the capsule network to which dynamic routing is not applied. The capsule vectors are used as local features. Finally, the similarity is evaluated by using the Manhattan distance for the global and local features of the two sentences. All neural networks, except self-attention, have a Siamese network structure.

The contributions of this study are as follows:

- To evaluate the semantic similarity of sentence pairs, we propose a model that uses global features, entire sentence information, and local features, localized sentence information, simultaneously. The proposed model can adjust whether to focus more on global information or local information, and it is seen that the accuracy is higher than the existing models that use only global information.
- In this study, the effect of dynamic routing on similarity evaluation was investigated. Since the similarity of the meaning of sentences is relatively free in the positioning of the corresponding phrases, it was found that dynamic routing hindered the correct evaluation. In addition, experiments were conducted on both English and Korean datasets to prove the language independency of the model proposed.

This paper is composed as follows. Section 2 briefly describes the models of former studies and our proposed model in evaluating sentence similarity. Section 3 minutely describes the proposed model for global and local feature extraction. Section 4 describes the data used in this study, the hyperparameters used in the experiment, and the experimental results. Section 5 discusses the results. Finally, Section 6 discusses the summary of this study and proposes future studies.

2. Related Works

Sentence similarity evaluation is used in various fields, and recently, it has shown good performance by utilizing deep learning [2–4]. Ref. [2] evaluates the similarity by applying the Siamese network structure to LSTM [13], a family of recurrent neural networks (RNN) that perform well in sequential data. The Siamese network is a structure in which two inputs are entered simultaneously into a single neural network [2]. Words constituting sentences are represented as vectors through word embedding. The two sentences composed of word vectors are entered into the LSTM and learning proceeds. The similarity is evaluated using Manhattan distance calculation, which is $\exp(-\|h_L^l - h_L^r\|_1)$, where h_L^l and h_L^r are the output vectors of the last hidden states representing each sentence.

Ref. [3] evaluates the similarity by applying the Siamese network structure to CNN and LSTM. Ref. [3] converts two sentences into two vectors. Then, local features are extracted through CNN, which extracts information on adjacent words. The extracted local features are entered as an LSTM, and LSTM is trained. The similarity between the two sentences is calculated by applying the Manhattan distance to the LSTM output vectors.

Ref. [4] evaluates the similarity by applying the Siamese network structure to group CNN and bidirectional GRU (Bi-GRU), a family of RNNs. [4] converts two sentences into two vectors. Then, multi-local features are extracted through group CNN, and the most representative local features are obtained by applying max-pooling to the multi-local features. These extracted local features are concatenated with the vector, then it enters the input of Bi-GRU. The similarity is determined by applying Manhattan distance to the Bi-GRU output vectors.

To consider whole sentence information, our proposed model in this study uses the last hidden state as a global feature by applying the Siamese network structure to Bi-LSTM. Additionally, to use localized sentence information representing contextual information, after applying self-attention the feature extracted from the capsule network is used as a local feature. Finally, sentence similarity is evaluated by applying Manhattan distance to global and local features.

3. Materials and Methods

Figure 1 shows the overall structure of the proposed model including word embedding, Bi-LSTM, self-attention, and a capsule network. In this model, global features are extracted from Bi-LSTM and local features from the capsule network. Global and local features are used to evaluate semantic similarity using the Manhattan distance.

Figure 1. Model of global and local features extraction for semantic similarity evaluation.

All neural networks, except self-attention, have the Siamese network structure. The Siamese network is a structure in which two input values enter a single neural network, sharing the weight of the neural network. Features created through the trained Siamese network generate vectors with short distances if the input sentences are similar, and vectors with long distance if they are not similar [2–4].

3.1. Word Embedding

In the field of natural language processing, words constituting text are expressed as vectors through word embedding [2–4]. Word embedding refers to expressing a word's meaning as dense vectors so that computers can understand human words [17,18]. This stems from the assumption in distributional semantics that words appearing in similar

contexts have similar meanings. Based on this, words with similar meanings are expressed in similar vectors [19].

In this study, Word2Vec [17,18], one of the word-embedding techniques, is used to express words as vectors. Word2Vec is a method to vectorize words using the target word and its surrounding words in a predetermined window size. Word2Vec updates learning weights by maximizing the dot product of the target word and its surrounding words. In this way, words are expressed as vectors representing their meanings.

3.2. Bidirectional Long Short-Term Memory

RNN is a neural network with good performance for sequential data processing, but learning ability is poor due to gradient vanishing, in which the gradient disappears during backpropagation, as the timestep, which refers to the order of input data, increases [7]. This problem is called long-term dependency. To solve the shortcomings of RNN, LSTM has been proposed. LSTM has a structure in which a cell state (c_t) that can have long-term memory is added to the hidden state (h_t) of RNN and solves long-term dependency [7]. To obtain h_t and c_t , LSTM uses an input gate (i_t), a forget gate (f_t), an output gate (o_t), and a cell (g_t) that analyzes h_{t-1} of the previous timestep and (x_t) of the input information of the current timestep. f_t is a gate that determines how much information of c_{t-1} is preserved by calculating the Hadamard product of f_t and c_{t-1} . i_t is a gate that determines how much information of x_t is also reflected by calculating the Hadamard product of i_t and g_t . o_t is a gate that determines h_t , the current hidden state, by calculating the Hadamard product of o_t and $\tanh(c_t)$. The equations of LSTM are as follows:

In Equation (1), the sigmoid function in i_t , f_t , and o_t results in adjusting how much of that information will be reflected. W_{xi} , W_{xf} , W_{xo} , and W_{xg} are the learning weight matrices connected to x_t , W_{hi} , W_{hf} , W_{ho} , and W_{hg} are connected to h_{t-1} , and b_i , b_f , b_o , and b_g are biases of the corresponding layers. c_t is updated by adding the Hadamard product of f_t and c_{t-1} and the Hadamard product of i_t and g_t . Finally, h_t is created by Hadamard product of $\tanh(c_t)$ and o_t .

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ y_t = h_t &= o_t \odot \tanh(c_t) \end{aligned} \quad (1)$$

Bi-LSTM has a structure with a forward LSTM that sequentially approaches from x_1 to x_L and a backward LSTM that sequentially approaches from x_L to x_1 . Hidden states sequentially processed in Bi-LSTM are defined as follows:

Equation (2) is the concatenation of the hidden state of the forward LSTM of timestep i and the hidden state of the backward LSTM, where L refers to the length of the sequence. In Equation (3), H is the concatenation of the hidden state of all timesteps. This H is then inputted to the attention mechanism.

$$h_i = [LSTM(x_i^f); LSTM(x_L^b)] \quad (2)$$

$$H = [h_1, h_2, \dots, h_{L-1}, h_L] \quad (3)$$

The global features that contain the information of the whole sentence are defined as follows:

Equation (4) refers to the concatenation of the last hidden state of Bi-LSTM. The global features are used with local features, to be mentioned in the section on the capsule network, to further improve the accuracy of the model in the future.

$$h_L = [LSTM(h_L^f); LSTM(h_1^b)] \quad (4)$$

3.3. Attention Mechanism

The attention mechanism [8,14–16] is a method of correlating words in sentences. In this study, self-attention using only one sentence is used, and H , which is the hidden state of Bi-LSTM, is used as an input value. Self-attention is defined as follows:

In Equation (5), h_i is the hidden state of the current timestep, and h_j is the hidden state of any timestep including the current one. W_i and W_j refer to the learning weights of the corresponding timesteps i and j , and b_i is the bias vector. In Equation (6), W_a is a learning weight that calculates the importance of each word in terms of the current word. e_{ij} in Equation (6) is a scalar value representing the importance of h_j in terms of h_i . The importance of words is normalized to a probability value by Equation (7). Then, c_i containing context information is extracted by Equation (8). By multiplying the hidden value of each word by the importance probability scalar and adding all the results, the final importance vector of a given word can be obtained. C is a matrix of $\mathbb{R}^{L \times U}$ when the length of the sentence is L and the number of units of Bi-LSTM is U .

$$h_{ij} = \tanh(W_i h_i + W_j h_j + b_i) \quad (5)$$

$$e_{ij} = \tanh(W_a h_{ij} + b_a) \quad (6)$$

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_k^L \exp(e_{ik})} \quad (7)$$

$$c_i = \sum_j^L a_{ij} h_j \quad (8)$$

3.4. Capsule Network

In this study, a capsule network using CNN is used to extract local features [10–13]. The capsule network consists of two CNNs, *Conv1* and *Conv2*. In this study, c extracted from self-attention is used as the input value of the capsule network. *Conv1* proceeds as follows:

In Equation (9), $f(\cdot)$ refers to the activation function, i refers to the index of the word, and h refers to the kernel size. w_k refers to a learning weight for the convolution filter and has a matrix of $\mathbb{R}^{h \times U}$, and b_k refers to the bias vector. *Conv1* is a typical CNN, extracting combined information of adjacent words as much as h does. Through this, features for *Conv1* are created. Finally, *Conv1* has a matrix of $\mathbb{R}^{(L-h+1) \times N}$ when the length of the sentence is L and the number of filters that increase the amount of learning is N .

$$\text{Conv1}_i = f\left(w_k * c_{i-\frac{h-1}{2}:i+\frac{h-1}{2}} + b_k\right) \quad (9)$$

Conv2 uses the features created in *Conv1* as an input value. *Conv2* is processed by Equation (9), like *Conv1*, but it uses the filter size corresponding to the entire size of input values to subdivide the expression of the entire sentence [13,20]. The features of *Conv2* extracted in this way have a matrix of $\mathbb{R}^{1 \times N}$ by the number of filters N . After that, PrimaryCaps with $\mathbb{R}^{D \times \frac{N}{D}}$ dimension is created by reconstructing the number of dimensions of the *Conv2* function. Here, PrimaryCaps means capsules that have subdivided the entire sentence information into D pieces. To normalize the size of the vector, PrimaryCaps uses squash, a nonlinear function [10–13]. Squash is defined as follows:

In Equation (10), P_j means the one capsule. In this study, PrimaryCaps with squash applied are used as local features.

$$v_j = \frac{\|P_j\|^2}{1 + \|P_j\|^2} \frac{P_j}{\|P_j\|} \quad (10)$$

3.5. Similarity Measure

Global features extracted from Bi-LSTM and local features extracted from the capsule network are generated for each of the two sentences. In this study, the similarity between two sentences is evaluated by applying the Manhattan distance for global features and local features. The Manhattan distance is close to 1 if both vectors are similar and close to 0 if they are dissimilar [2–4]. The similarity between global and local features is defined as follows:

In Equations (11) and (12), l and r mean two sentences. In Equation (11), h_L is a global feature using the last hidden state of Bi-LSTM. In Equation (12), P means PrimaryCaps, and D means the number of dimensions of PrimaryCaps.

$$Sim_G(h_L^l, h_L^r) = \exp(-\|h_L^l - h_L^r\|_1) \quad (11)$$

$$Sim_L(P^l, P^r) = \frac{1}{D} \sum_i^D \exp(-\|P_i^l - P_i^r\|_1) \quad (12)$$

The final similarity value is calculated by the alpha weight (α) as follows:

In Equation (13), α is a weight that can adjust which information to focus on among global features and local features. The value of α is determined experimentally. Finally, the predicted *Similarity* of 0.5 or higher determines similarity, and the others determine to non-similarity.

$$Similarity = \alpha Sim_G(h_L^l, h_L^r) + (1 - \alpha) Sim_L(P^l, P^r) \quad (13)$$

4. Experiments

In the experiment, the improved accuracy of the model proposed in this study is compared with that of the past models. Experiments are conducted in English and Korean to show that our model develops accuracy regardless of languages. To directly check the effect of dynamic routing, we have created models with or without dynamic routing.

4.1. Dataset

In this experiment, a corpus for learning representation of words and a corpus for learning similarity of given sentences are also needed. In order to learn the representations of English words, we use the Google News corpus [21]. For Korean, we use the raw Korean sentence corpus produced and distributed by Kookmin University in Korea [22]. Individual words are embedded by Word2Vec. These corpora have shown good results in existing sentence similarity evaluation research [2,3,23]. English and Korean words are represented as 300-dimensional vectors, and when the size of vocabulary appearing in a sentence is L , they have a matrix of $\mathbb{R}^{L \times 300}$. The next two subsections show the status of two datasets used for learning similarity evaluation models in English and Korean, respectively.

4.1.1. English Dataset

The English dataset used in this study is Quora Question Pairs [24]. A pre-processing is carried out in which the stopword and all special characters are deleted from all sentences, and upper case letters are changed to lower case letters. Besides, since the sentences could have a very short length after preprocessing, we collected sentences with more than five words. Table 1 is an example of English sentence pairs and their labels collected for the experiment.

In Table 1's label, 0 means that sentence pairs do not have similar meanings, and label 1 means that the pairs are similar in meaning. The number of data items for label 0 is 50,000, and the number of data items for label 1 is also 50,000, so a total of 100,000 data are used in this study. Thereafter, the ratio of the training set: validation set: test set is set to 8:1:1.

Table 1. Example of English sentence pairs and labels.

Sentence 1	Sentence 2	Label
I'm a 19-year-old. How can I improve my skills or what should I do to become an entrepreneur in the next few years?	I am a 19 years old guy. How can I become a billionaire in the next 10 years?	0
What are the life lessons that Batman teaches us?	What are the life lessons you can learn from the dark knight?	1

4.1.2. Korean Dataset

The Korean dataset used in this study is collected through the translation of Quora Question Pairs [24] by Google translator [25], Naver question pairs [26], Exobrain Korean paraphrase corpus [27], and German translation pairs developed by Hankuk University of Foreign Studies [28]. To use the elaborate Korean dataset, Naver Korean spell checker [29] is used. The experiment is conducted after dividing words into morphemes which are the smallest unit of words with meaning by using Kkma, a Korean morphological analyzer [30]. Table 2 is an example of Korean sentence pairs and labels collected for the experiment.

Table 2. Example of Korean sentence pairs and labels.

Language	Sentence 1	Sentence 2	Label
Korean	참고 인내하며 때를 기다려야 하는 날입니다.	일이 잘 풀릴 것 같은 날입니다.	0
English	It is a day when you have to be patient and wait for the time.	It's a day that seems to be going well.	
Korean	가장 저렴한 방법으로 치아 미백 효과를 낼 수 있는 방법은?	치아를 미백할 수 있는 저렴하고 효율적인 방법은?	1
English	What is the cheapest way to produce teeth whitening effect?	What is an inexpensive and efficient way to whiten teeth?	

In Table 2, in English is a pair of sentences translated from Korean, and the label of the sentence pair is marked as 1 if the meaning is similar and 0 if not similar. The number of individual data for label 0 is 5500, and the number of data items for label 1 is 5500, so a total of 11,000 data are used in this study. Thereafter, the ratio of the training set: validation set: test set is set to 8:1:1.

4.2. Hyperparameters

Table 3 shows the hyperparameters for each neural network used in the experiment.

Table 3. Hyperparameters used in the experiment.

Features	Description	Hyperparameter
Global	The number of Units of bidirectional long-short term memory (Bi-LSTM)	256
	Activation function used in the <i>Conv1</i>	Rectified Linear Unit
	Kernel size used in the <i>Conv1</i>	3
Local	The number of kernels used in the <i>Conv1</i>	256
	Activation function used in the <i>Conv2</i>	None
	Kernel size used in the <i>Conv2</i>	The total size of the Input
	The number of kernels used in the <i>Conv2</i>	
	The number of dimensions used in the PrimaryCaps	8

4.3. Accuracy Comparison According to α

Figure 2a,b show the performance comparison according to α of Equation (13) in Section 3.5. Figure 2a shows the accuracy of the English dataset and Figure 2b shows the

accuracy of the Korean. In the case of English, it can be seen that the higher the proportion of global features, the higher the accuracy. Its peak is at 80% of the weight of the global features. In Korea, accuracy also increases according to the weight of global features. However, the accuracy shows the peaks when the weight of the two features is around 5:5. This is because the proportion of local features is relatively higher in the case of Korean, with more free word order than English.

Figure 2. (a) English dataset accuracy according to α ; (b) Korean dataset accuracy according to α .

4.4. Result

Table 4 shows the accuracy of various models, including the previously proposed semantic similarity evaluation models. Acc shows the average accuracy of 10 experiments using Manhattan distance. Here, English Acc means the accuracy using the English dataset, and Korean Acc means accuracy using the Korean dataset.

Table 4. Model accuracy comparison according to experiments.

No.	Model	English Acc	Korean Acc
1	LSTM [2]	80.99%	87.25%
2	Convolutional neural networks (CNN) + LSTM [3]	80.44%	83.76%
3	Group CNN + bidirectional gated recurrent unit (Bi-GRU) [4]	82.18%	89.52%
4	Bi-LSTM	82.9%	90.28%
5	CNN + Max-Pooling	80.58%	88.26%
6	Capsule Network	81.57%	88.85%
7	Capsule Network + Dynamic Routing	78.82%	86.48%
8	LSTM + Capsule Network	82.07%	90.25%
9	Bi-LSTM + Capsule Network	82.15%	90.56%
10	LSTM + Self-Attention + Capsule Network	82.78%	90.61%
11	Bi-LSTM + Self-Attention + Capsule Network	82.9%	91.41%
12	Proposed Model	83.51%	92.08%

In Table 4, No. 1 to No. 4 are models using only global features. No. 1 to No. 3 are models proposed in the previous studies [2–4], and No. 4 and No. 5 are models using simple Bi-LSTM and CNN, respectively. These models are arbitrarily built to compare the accuracy of the RNN series model and the CNN series model. RNN shows higher accuracy than CNN in this task. No. 6 to No. 11 are models using only local features, i.e., the capsule network. No. 6 and No. 7 are experiments to verify the effect of dynamic routing when using a capsule network. In both English and Korean, it can be seen that dynamic routing causes performance degradation. No. 8 to No. 11 are models applying the capsule network to RNN-related models, but they do not use the last hidden state of LSTM. No. 10 and

No. 11 are experiments in which self-attention is added to No. 8 and No. 9, respectively. Finally, No. 12 is the proposed model using both global and local features.

5. Discussion

Existing studies [2–4] have tried to estimate the similarity of sentences mainly by using global features. We were able to improve accuracy by up to 8.32%p (Korean) by also utilizing local features. In addition, it was shown that an accuracy of up to 3.03%p can be achieved by simply changing the LSTM to Bi-LSTM (compare No. 1 to No. 4). A model capturing forward information and backward information simultaneously is more effective. Comparing No. 1 and No. 2, and No. 3 and No. 4, it can be seen that simply and mechanically combining the global features and the local features does not help improve performance. Rather, it only decreases the accuracy.

A capsule network is designed to solve the problem of CNN's pooling method, but rather CNN's pooling shows higher accuracy not only for the English dataset but also for the Korean dataset (see No. 5 and No. 7. in Table 4). Furthermore, we can confirm that the capsule network model, No. 6, which does not use dynamic routing, has higher accuracy than the CNN based models, No. 5 and No. 7. Through this, it can be confirmed that dynamic routing considering the spatial relationship of local features is inappropriate for use in the task of semantic similarity estimation. Unlike images in which the relative positions of the pixel chunks are fixed to some extent, the position of words or phrases in a sentence are relatively free.

As shown from No. 8 to No. 11, which are RNN-related models, it can be confirmed that the accuracy of the capsule network model with self-attention applied to Bi-LSTM is the best. In No. 4, No. 9, and No. 11, in Korean the degree of accuracy improvement by the capsule network and the self-attention is higher than in English. The semantics of Korean sentences is concentrated in a more local area than in English. In order to grasp the entire meaning of a Korean sentence, it is important to accurately understand the local meanings.

No. 12 is a proposed model in which α of Equation (13) is 0.8 in English and 0.5 in Korean. This can be seen that α of the Korean dataset concerning global features is lower than α of the English dataset. From the fact that the larger the alpha value, the higher the proportion of global features, we can infer that Korean depends more on local features since it has a freer order of words than English.

6. Conclusions

In this study, the semantic similarity of sentence pairs is evaluated using global and local sentence information. The proposed model consists of Bi-LSTM, self-attention, and capsule network, and all neural networks, except self-attention, apply the Siamese network. This model extracts global features through Bi-LSTM and extracts local features through the capsule network. The extracted global and local features are used when evaluating the semantic similarity of sentence pairs using Manhattan distance. This allows not only global and local features to be considered together but also the adjustment of which information to focus more closely on by α . As a result of comparing the existing models, models using only global features, models using only local features, and the model using both global and local features at the same time, it can be seen that the accuracy of the model using both global and local features at the same time is higher.

However, this study has a limitation. Even though α plays a very important role in determining the weight of global and local features, we have not been able to come up with a universal methodology for obtaining α . We will find a way to integrate this into the entire network to obtain the optimal α through learning. We will also apply another language model instead of Word2Vec to obtain more sophisticated sentence pair representations, and update the current network structure after analyzing the strengths and weaknesses to further improve the accuracy of the model.

Author Contributions: Conceptualization, T.-S.H. and Y.-S.K.; Data curation, T.-S.H.; Formal analysis, T.-S.H., J.-D.K. and C.-Y.P.; Methodology, T.-S.H.; Project administration, Y.-S.K.; Resources, T.-S.H.; Software, T.-S.H.; Supervision, Y.-S.K.; Validation, J.-D.K. and C.-Y.P.; Writing—original draft, T.-S.H.; Writing—review and editing, Y.-S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C2006010).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <https://code.google.com/archive/p/word2vec/>; <http://nlp.kookmin.ac.kr>; <https://www.kaggle.com/quora/question-pairs-dataset>; <http://kin.naver.com/>; http://aiopen.etri.re.kr/service_dataset.php.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Marelli, M.; Menini, S.; Baroni, M.; Bentivogli, L.; Bernardi, R.; Zamparelli, R. A sick cure for the evaluation of compositional distributional semantic models. In Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 26–31 May 2014; pp. 216–223.
- Mueller, J.; Thyagarajan, A. Siamese recurrent architectures for learning sentence similarity. In Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16), Phoenix, AZ, USA, 12–17 February 2016; pp. 2786–2792.
- Pontes, E.L.; Huet, S.; Linhares, A.C.; Torres-Moreno, J.M. Predicting the semantic textual similarity with siamese CNN and LSTM. *arXiv* **2018**, arXiv:1810.10641.
- Li, Y.; Zhou, D.; Zhao, W. Combining Local and Global Features into a Siamese Network for Sentence Similarity. *IEEE Access* **2020**, *8*, 75437–75447. [\[CrossRef\]](#)
- Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318.
- Banerjee, S.; Lavie, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Proceedings of the acl Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; pp. 65–72.
- Lin, Z.; Feng, M.; Santos, C.N.D.; Yu, M.; Xiang, B.; Zhou, B.; Bengio, Y. A Structured Self-Attentive Sentence Embedding. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Palais des Congrès Neptune, Toulon, France, 24–26 March 2017.
- Ma, Q.; Yu, L.; Tian, S.; Chen, E.; Ng, W.W. Global-Local Mutual Attention Model for Text Classification. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2019**, *27*, 2127–2139. [\[CrossRef\]](#)
- Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS-2017), Long Beach, CA, USA, 4–9 December 2017; pp. 3859–3869.
- Yang, M.; Zhao, W.; Chen, L.; Qu, Q.; Zhao, Z.; Shen, Y. Investigating the transferring capability of capsule networks for text classification. *Neural Netw.* **2019**, *118*, 247–261. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wu, Y.; Li, J.; Wu, J.; Chang, J. Siamese capsule networks with global and local features for text classification. *Neurocomputing* **2020**, *390*, 88–98. [\[CrossRef\]](#)
- Kim, J.; Jang, S.; Park, E.; Choi, S. Text classification using capsules. *Neurocomputing* **2020**, *376*, 214–221. [\[CrossRef\]](#)
- Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
- Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#) [\[PubMed\]](#)
- Gong, J.; Qiu, X.; Wang, S.; Huang, X. Information aggregation via dynamic routing for sequence encoding. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–25 August 2018; pp. 2742–2752.
- Zheng, G.; Mukherjee, S.; Dong, X.L.; Li, F. Opentag: Open attribute value extraction from product profiles. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 22 July 2018; pp. 1049–1058.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 12–15 December 2013; pp. 3111–3119.
- Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
- Sahlgren, M. The distributional hypothesis. *Ital. J. Disabil. Stud.* **2008**, *20*, 33–53.

20. Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large Kernel Matters—Improve Semantic Segmentation by Global Convolutional Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4353–4361.
21. Google News. Available online: <https://code.google.com/archive/p/word2vec/> (accessed on 7 September 2020).
22. Korean Raw Corpus. Available online: <http://nlp.kookmin.ac.kr> (accessed on 7 September 2020).
23. Song, H.J.; Heo, T.S.; Kim, J.D.; Park, C.Y.; Kim, Y.S. Sentence similarity evaluation using Sent2Vec and siamese neural network with parallel structure. *J. Intell. Fuzzy Syst.* **2021**, Pre-press.
24. Quora Question Pairs. Available online: <https://www.kaggle.com/quora/question-pairs-dataset> (accessed on 7 September 2020).
25. Google Translator. Available online: <https://translate.google.com/> (accessed on 2 February 2021).
26. Naver Question Pairs. Available online: <http://kin.naver.com/> (accessed on 7 September 2020).
27. Exobrain Korean Paraphrase Corpus. Available online: http://aiopen.etri.re.kr/service_dataset.php (accessed on 7 September 2020).
28. German Translation Pair of Hankuk University of Foreign Studies. Available online: <http://deutsch.hufs.ac.kr/> (accessed on 7 September 2020).
29. Naver Spell Checker. Available online: https://search.naver.com/search.naver?sm=top_hy&fbm=0&ie=utf8&query=%EB%84%A4%EC%9D%B4%EB%B2%84+%EB%A7%9E%EC%B6%A4%EB%B2%95+%EA%B2%80%EC%82%AC%EA%B8%B0 (accessed on 7 September 2020).
30. Kkma Morpheme Analyzer. Available online: <http://kkma.snu.ac.kr/> (accessed on 7 September 2020).