Information Sciences 541 (2020) 123–135

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Paraphrase thought: Sentence embedding module imitating human language recognition

Myeongjun Jang [a], Pilsung Kang [b,*]

[a] AI Language Labs, SK Telecom, Seoul, Republic of Korea
[b] School of Industrial Management Engineering, Korea University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 12 November 2018
Received in revised form 2 April 2020
Accepted 30 May 2020
Available online 2 July 2020

Keywords:
Sentence embedding
Recurrent neural network
Paraphrase
Semantic coherence
Natural language processing

ABSTRACT

Sentence embedding is an important research topic in natural language processing. It is essential to generate a good embedding vector that fully reflects the semantic meaning of a sentence in order to achieve an enhanced performance for various natural language processing tasks, such as machine translation and document classification. Thus far, various sentence embedding models have been proposed, and their feasibility has been demonstrated through good performances on tasks following embedding, such as sentiment analysis and sentence classification. However, because the performances of sentence classification and sentiment analysis can be enhanced by using a simple sentence representation method, it is not sufficient to claim that these models fully reflect the meanings of sentences based on good performances for such tasks. In this paper, inspired by human language recognition, we propose the following concept of semantic coherence, which should be satisfied for a good sentence embedding method: similar sentences should be located close to each other in the embedding space. Then, we propose the Paraphrase-Thought (P-thought) model to pursue semantic coherence as much as possible. Experimental results on three paraphrase identification datasets (MS COCO, STS benchmark, SICK) show that the P-thought models outperform the benchmarked sentence embedding methods.

1. Introduction

Sentence embedding, which transforms sentences into low-dimensional vector values reflecting their meanings, is a highly important task in natural language processing (NLP). By mapping unstructured text data into a certain form of structured representation, the embedding vector can enhance the performances of various NLP tasks, such as machine translation [1,2], document classification [3,4], and sentence matching [5]. As sentence embedding plays an import role in NLP, various methods [6–11] have been proposed since the advent of the Doc2vec method [12]. Typically, these methods exhibit better performances than benchmarked embedding methods for common NLP tasks, such as document classification or sentiment analysis. However, this is not a direct evaluation of how well semantic meanings are preserved by the proposed embedding method.

Indirect methods for evaluating sentence embedding are not sufficient to evaluate the main property of sentence embedding techniques, i.e., how well semantic relationships between sentences are preserved. Iyyer et al. [13] showed that it is possible to achieve a fairly good performance in document classification using a simple document representation vector, i.e., an average of word vectors in the document. Even for classic document representation methods, in which word

* Corresponding author at: 801A Innovation Hall, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul, Republic of Korea.
E-mail address: pilsung_kang@korea.ac.kr (P. Kang).

https://doi.org/10.1016/j.ins.2020.05.129
0020-0255/

sequences or semantic relationships between words are not considered, e.g., bag of words (BoW) or term frequency-inverse document frequency (TF-IDF), highly accurate classification results can be achieved using a Naïve Bayesian classifier [14]. This means that a good performance on a classification task can be achieved without the use of embedding vectors. In other words, a good classification performance for common NLP tasks using a certain type of sentence embedding method does not guarantee that the embedding method can successfully preserve the semantic relationship between sentences.

In this paper, in order to overcome the limitations of indirect sentence embedding evaluation strategies, we propose the following concept of semantic coherence, which should be satisfied by a good sentence embedding method: sentences having similar meanings should be placed close to each other in the embedding space. Then, we propose a new sentence embedding model named Paraphrase-Thought (P-thought), which can maximally pursue semantic coherence during training. The P-thought model is designed as a dual generation model, which receives a single sentence as input and generates both the input sentence and its paraphrase sentence simultaneously. The proposed P-thought model is evaluated through a task of measuring the semantic coherence, the STS Benchmark task and the semantic relatedness task. Experimental results show that the proposed P-thought model yields a better performance than benchmarked models in most of the tasks.

The remainder of this paper is organized as follows. In Section 2, we briefly review previous research on sentence embedding. In Section 3, we propose the concept of semantic coherence and a new metric: paraphrase coherence (P-coherence). In Section 4, we describe the structure of the P-thought model. In Section 5, experimental settings are described for each task, followed by results and discussions. In Section 6, we conclude the present work with some discussion of future research directions.

2. Related work

Recent work on sentence embedding ranges from simple extensions of the word embedding vector [12,9,7,11,15] to neural network models specialized for handling a sequence of words appearing in a sentence [6,10]. The Distributed Bag of Words version of Paragraph Vector (PV-DBOW) and Distributed Memory Model of Paragraph Vectors (PV-DM) methods, which were proposed in Doc2vec [12], learn sentence vectors based on the same principle: maximizing the probability to predict words in the same sentence. Arora et al. [9] proposed a model that computes the sentence embedding vector as a weighted average of word embedding vectors in a sentence. By re-weighting the weights of words in a sentence, the authors achieved an improved performance in textual similarity tasks, and outperformed a complex model based on recurrent neural network (RNN). Unlike in Doc2vec [12], in Doc2vecC [11], the sentence embedding vector is defined as a simple average of word embedding vectors. The idea behind doc2vecC, i.e., using an average of word embedding vectors to represent the global context of the sentence, had already been proposed by Huang et al. [16]. In addition, doc2vecC applies a corruption mechanism that randomly removes words from a sentence and generates a sentence embedding vector with the remaining words. This simple idea significantly reduced the total amount of training time. Similar to previous methods, Sent2vec [7] defines the sentence vector as an average of word embedding vectors. However, unlike other models using word embedding vectors of single words (i.e., uni-gram), it considers n-gram vectors in addition to uni-gram vectors when training the sentence embedding model.

The Skip-thought model [6], which has a sequence to sequence (Seq2Seq) structure, is an extension of the Skip-gram [17] model, where the basic unit for network learning is a sentence instead of a word. Similar to the Skip-gram model, which learns word embedding vectors by training the network to predict the surrounding words when the center word is given, Skip-thought is trained to encode the input sentence and generate its preceding and following sentences. By using the generated sentence vectors as the input of a simple linear model, Skip-thought exhibited an improved performance for document classification and sentiment analysis. Inspired by previous results in computer vision, where many models are pretrained based on ImageNet [18], Conneau et al. [10] conducted research on whether supervised learning tasks are helpful for learning sentence embedding vectors. Through experiments, Conneau et al. [10] claimed that sentence embedding vectors generated from a model that is trained based on a natural language inference (NLI) task yield a state-of-the-art performance when leveraged in other NLP tasks. In particular, they found that a model with a bi-directional Long-short term memory (LSTM) structure and max pooling trained on the Stanford Natural Language Inference (SNLI) dataset [19], named InferSent, exhibited the best performance. Jeong et al. [20] developed a sentence similarity identification model inspired by the image association of human behavior. Their essential idea is that humans will associate paraphrase sentences to a similar image. They employed the Siamese network [21] to represent two sentences to vectors simultaneously and used the Text2image GAN network [22] as a subnetwork inside the Siamese network. To train the network, they added a normalized-distance based contrastive loss to the original Text2image network loss function. As a result, the model is trained to minimize the distance between similar sentences and maximize that of dissimilar sentences. The model is trained based on the MS-COCO [23] caption and image dataset.

3. Semantic coherence

3.1. Defining semantic coherence

Although two sentences may employ different words or different structures, people will recognize them as the same sentence as long as the implied semantic meanings are highly similar. Consider the following two sentences:

- **Sentence 1:** Jang was caught by professor Kang while playing the computer game in the lab.
- **Sentence 2:** Professor Kang came to the lab and witnessed Jang playing the computer game.

Although these two sentences exhibit a clear difference with respect to both the sentence structure and word usage, people can immediately perceive that they convey the same meaning. Hence, a good sentence embedding approach should satisfy the property that if two sentences have different structures but convey the same meaning (i.e., paraphrase sentences), then they should have the same, or at least similar, embedding vectors. Based on this, we define semantic coherence as follows.

Definition 1. The degree of semantic coherence of a sentence embedding model is proportional to the similarity between the representation vectors of paraphrase sentences generated by the model.

If the representation vectors of paraphrase sentences are located close to each other in the embedding space, this implies that there is little difference between their vector values. Thus, when the representation vector value of a sentence is given, it should be possible to generate the given sentence and its paraphrase sentences. Consequently, we can derive the following hypothesis.

Hypothesis 1. If it is possible to generate an input sentence and its paraphrase sentence simultaneously from the vector value of the input sentence, then the sentence embedding model can enhance the semantic coherence.

In this study, we propose a new sentence embedding model to satisfy the above hypothesis.

3.2. Evaluating semantic coherence: paraphrase coherence

To evaluate semantic coherence, we should measure the densities of paraphrase sentences. This requires multiple pairs of paraphrase sentences that share the same meaning. Thus, previous metrics that simply calculate the matching degree of two sentences are insufficient.

In this study, inspired by topic coherence, which is used to determine the optimal number of topics in topic modeling, we propose a new evaluation metric called paraphrase coherence (P-coherence) to measure the semantic coherence. Topic coherence measures how effectively the highly weighted top k words of a topic satisfy coherence [24]; topic coherence is computed as follows:

$$Topic-coherence = \sum_{i<j} Score(w_i, w_j), \tag{1}$$

where w_i and w_j are the top i^{th} and j^{th} words of the same topic, respectively. Although various methods exist to define the score between two words [25], we adopted the idea of the pointwise mutual information (PMI) measure [24], defined as follows:

$$Score_{PMI}(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}, \tag{2}$$

where $p(w_i, w_j)$ is the probability of words w_i and w_j appearing together in a randomly selected document, and $p(w_i)$ and $p(w_j)$ are the marginal probabilities that the words w_i and w_j appear in the randomly selected document, respectively.

Unlike in topic coherence, which defines the probability of two words appearing together based on a simple word count, we should consider the relationship between two sentences by leveraging their representation vector values. Hence, we replace the co-occurrence probability $p(w_i, w_j)$ in topic coherence with the dot product of two sentence representation vectors because the dot product of two vectors is widely used as an unnormalized probability in many studies [26,27]. Next, we replace the marginal probability for word occurrence in topic coherence with the L_2-norm of the sentence vector, derived from the dot product. As a result, the score between two sentences takes a value between -1 and 1: the higher the score value, the stronger is the relationship between the two sentences. The equation representing the proposed score is as follows:

$$Score(sv_i, sv_j) = \frac{sv_i \cdot sv_j}{\|sv_i\| \|sv_j\|}, \tag{3}$$

where sv_i and sv_j are the representation vectors of the sentences i and j, respectively. Finally, P-coherence is defined as the average score of all pairs of paraphrase sentences:

$$\begin{aligned} &P-coherence(U_k) = Average\left(\frac{sv_i \cdot sv_j}{\|sv_i\| \|sv_j\|}\right), \\ &sv_i,\ sv_j \in U_k, \quad k = 1, \ldots, N, \end{aligned} \tag{4}$$

where U_k is the k^{th} set of paraphrase sentences, and N is the number of paraphrase sets. For instance, if there are four paraphrase sentences for each paraphrase set, then the P-coherence for each paraphrase set is calculated as the average score of ${}_4C_2 = 6$ sentence pairs. The total P-coherence is the average P-coherence for each paraphrase set:

$$P-coherence_{Total}=\frac{1}{N}\sum_{k=1}^{N}P-coherence(U_k). \tag{5}$$

4. Paraphrase thought

4.1. Model structure

Assume that a sentence tuple (s,p) is given, where p is the paraphrase sentence of the sentence s. Let x_t be the t^{th} word of the sentence s and y_t be the t^{th} word of the sentence p. To maximize the semantic coherence defined above, it should be possible to generate both the sentence itself and its paraphrase sentence from the representation vector of an input sentence. Therefore, the proposed P-thought model is designed as a dual generation model, which generates both s and p simultaneously when the sentence tuple (s,p) is given.

We employed an Seq2Seq structure with a gated recurrent unit (GRU) [28] cell for the P-thought model. The *encoder* transforms the sequence of words of an input sentence into a fixed-sized representation vector, whereas the *decoder* generates the target sentence based on the given sentence representation vector. The proposed P-thought model has two decoders. When the input sentence is given, the first decoder, named $auto-decoder$, generates the input sentence as it is. The second decoder, named $paraphrase-decoder$, generates the paraphrase sentence of the input sentence.

4.2. Objective

Similar to other sequence learning tasks in NLP, the purpose of the P-thought model is to minimize the negative log likelihoods of the two decoders. Furthermore, according to Hypothesis 1, the P-thought model should satisfy the condition that it can encode the sentence s and generate the sentences s and p simultaneously when the sentence pair (s,p) is given. This condition can be written as follows:

$$P(s)P(s|s;\theta_{ss})=P(s)P(p|s;\theta_{sp}), \tag{6}$$

where $P(s)$ is the marginal probability of the input sentence s, and θ_{ss} and θ_{sp} are the parameters of $auto-decoder$ and $paraphrase-decoder$, respectively. Thus, the problem can be formulated as the following multi-objective problem with a constraint:

$$\begin{aligned}&\textbf{Objective1}: \min_{\theta_{ss}} l_A(f(s;\theta_{ss}),s),\\ &\textbf{Objective2}: \min_{\theta_{sp}} l_P(f(s;\theta_{sp}),p),\\ &s.t \quad P(s)P(s|s;\theta_{ss})=P(s)P(p|s;\theta_{sp}),\end{aligned} \tag{7}$$

where l_A and l_P are the negative log likelihoods of $auto-decoder$ and $paraphrase-decoder$, respectively. In this case, the constraint term can be rewritten as follows:

$$-\log P(s|s;\theta_{ss})=-\log P(p|s;\theta_{sp}), \tag{8}$$

The left and right terms of transformed equation represent the negative log likelihoods of $auto-decoder$ and $paraphrase-decoder$, respectively. Hence, the constraint term can be written as follows:

$$l_A(f(s;\theta_{ss}),s)=l_P(f(s;\theta_{sp}),p). \tag{9}$$

By introducing the Lagrange multiplier, the multi-objective optimization problem is transformed into the following minimization problem:

$$\begin{aligned}\min L \quad &= l_A(f(s;\theta_{ss}),s)+l_P(f(s;\theta_{sp}),p)\\ &\quad -\lambda(l_A(f(s;\theta_{ss}),s)-l_P(f(s;\theta_{sp}),p))\\ &= (1-\lambda)l_A(f(s;\theta_{ss}),s)+(1+\lambda)l_P(f(s;\theta_{sp}),p)),\\ where \quad &\lambda\neq 0.\end{aligned} \tag{10}$$

In this case, a value of $\lambda>1$ or $\lambda<-1$ leads to maximizing the negative log likelihood of $auto-decoder$ and that of $paraphrase-decoder$, respectively. To avoid this problem, the allowable range for λ is set to $-1<\lambda<0$ or $0<\lambda<1$. However, it is desirable to set the appropriate λ-value to greater than 0, considering that auto decoding is trivial copying task which is much easier than paraphrase generation. Thus, the objective of the P-thought model is the sum of the negative

log likelihood of $auto-decoder$ with that of $paraphrase-decoder$ with a higher weight. The λ is an optimal value which should also be found during the optimization process. We used a grid search to find the λ since it is impossible to find the optimal value deterministically. The grid search result is described in B.

4.3. Vocabulary expansion

The number of unique words appearing in our training dataset is only about 35,000, which is considerably fewer than the number of words in the English language. This may be problematic, in that many words are treated as out of vocabulary after model training. To solve this problem, motivated by the idea of cross-lingual embedding [29], Skip-thought attempts to learn a matrix that maps the words of a pretrained word2vec model [17] to one of 20,000 words in their training dataset. However, this approach suffers from the problem that a word can be mapped to another word whose actual meaning is significantly different, only because it has a high similarity with the original word in the embedding space. For example, the word 'endogenous' was mapped to the word 'neuronal,' despite the semantic differences.

We extracted the vector values of words that appear in our training dataset from the pretrained Glove vector [30] to resolve the problem described above. In the pretrained Glove vectors, the semantic relationships between words are reflected in the geometrical structures between word vectors. Therefore, even when vector values of words that are unused during training occur, the information loss can be reduced because the geometric relationships between word vectors are well preserved if the model is sufficiently trained. By using this method, we are able to handle 2.1 million words without the effort of training an extra mapping matrix.

5. Experiments

5.1. Experimental settings

We used the captions of the MS-COCO dataset [23] to train the P-thought model. This dataset has been employed in various paraphrase generation studies [31,32]. The MS-COCO dataset has more than five captions for each image, which allows us to generate more than ${}_5P_2 = 20$ unique sentence pairs. For training, we used the 2014-Validation and 2017-Training datasets. Descriptions of these datasets are provided in Table 1. Simple tokenizing was performed as text preprocessing for the captions. Evaluating the quality of the sentence representation vector is demanding work because a globally accepted evaluation metric that can directly measure embedding quality does not exist. Therefore, we indirectly evaluated the quality of sentence embedding vectors by measuring the performance of several downstream tasks whose primary purpose is to preserve the semantic similarity between two given sentences.

We employed three different encoder structures, as shown in Fig. 1, to investigate the model performances according to different levels of model complexity. The first encoder structure has one layer with a bi-directional RNN (Bi-RNN). The sentence embedding vector of this encoder structure consists of the concatenated values of the final state values of the forward and backward RNN. The second encoder structure contains two layers, with only a forward RNN. The sentence embedding vector is generated by concatenating the final states of both layers. The third encoder structure contains two layers of Bi-RNN. The sentence embedding vector is generated from the concatenated values of the final states of the second layer's forward and backward RNNs. The overall structure of the P-thought model, including the decoder part, is illustrated in Fig. 2. These three models were trained under the same conditions. We fixed the length of the input sentence because excessively long sentences significantly increase the training time. Therefore, we set the maximum length of a sentence to 50 referring to the maximum length of input sentence used in previous research which using Seq2Seq structure [28,33,1]. The number of hidden units is set to 1,200, which results in 2,400-dimensional sentence embedding vectors after concatenation. We employed Xavier initialization [34], and gradient computations and weight updates were performed with a mini-batch size of 128. All models were trained for four epochs using the Adam optimizer [35].

5.2. P-coherence

To measure the P-coherence, we used the 2017-Validation dataset from the MS-COCO caption dataset, which has no overlap with the training dataset. A description of the dataset used for evaluating the P-coherence is provided in Table 2. We selected PV-DBOW, Skip- thought, SIF, Sent2vec, InferSent, and the model of Jeong et al. [20] as benchmark models. In

Table 1
Training data description.

	2014-Validation	2017-Training	Total
No. of unique images	40,504	118,284	123,287
No. of unique captions	202,654	591,753	593,968
No. of unique sentence pairs	811,426	2,368,926	2,467,293
No. of unique words	–	–	34,826

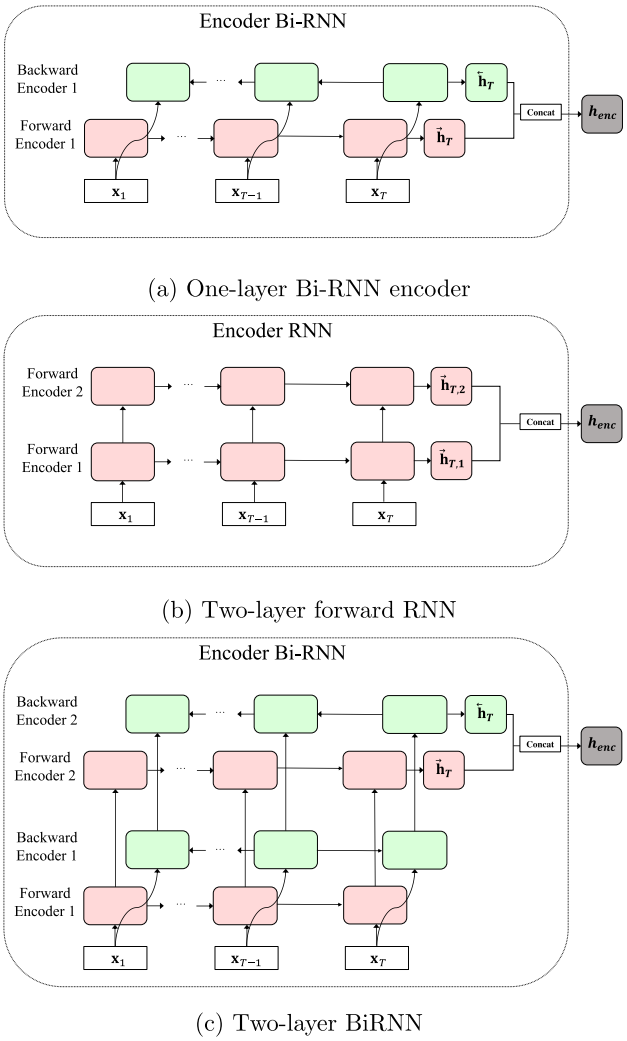


Fig. 1. Three encoder structures of the P-thought model.

the case of PV-DBOW, we employed the datasets used for both training P-thought and evaluating the P-coherence to learn the sentence vectors. For the remaining models, we used the publicly available pretrained models.

The experimental results are summarized in Table 3. It can be observed that the P-thought models with relatively complex encoder structures outperformed other benchmarked models. In the case of P-thought with a one-layer Bi-RNN, the P-coherence value is comparable to that of InferSent, and superior to the other benchmarked models. Among the benchmarked models, InferSent yielded a significantly higher P-coherence value than the other models, which implies that InferSent preserved the semantic coherence when learning the sentence representation vectors.

In addition to the quantitative evaluation provided in Table 3, we reduced the generated sentence vectors to two-dimensional vectors using *t*-SNE [36] and created scatter plots to qualitatively investigate how effectively the paraphrase sentences satisfied coherence. For the sake of visualization, we extracted the paraphrase sentences for five images and marked them with different colors and shapes. The extracted paraphrase sentences are presented in Table 4, and the scatter plots are given in Fig. 3. It can easily be observed that paraphrase sentence vectors learned by the models with high P-coherence values (P-thought and InferSent) are more concentrated than those of the other models.

5.3. STS Benchmark task

We carried out the STS Benchmark task [37] to evaluate how well the models preserve the meanings of sentences through a more generally conducted task. The dataset for this task consists of 8628 sentence pairs and corresponding human rated

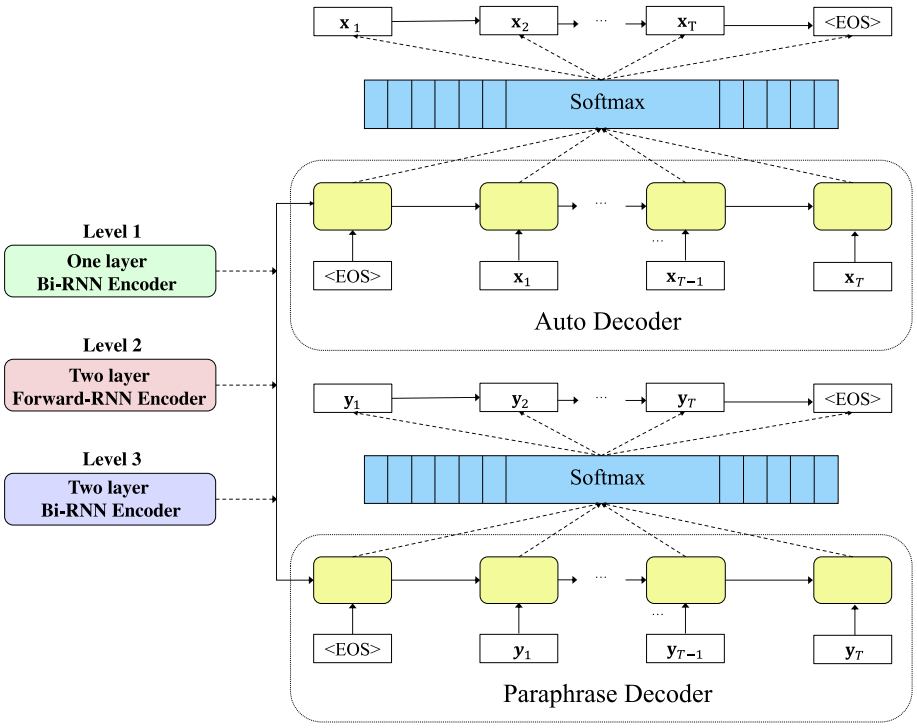


Fig. 2. P-thought model structure.

Table 2
2017-Validation dataset description.

No. of unique images	No. of unique captions	No. of unique sentence pair	No. of unique words
5000	25,014	100,142	8641

Table 3
Experimental results for evaluating the P-coherence.

Model	P-coherence
PV-DBOW (Le and Mikolov [12])	0.0099
Uni-skip (Kiros et al. [6])	0.5328
Bi-skip (Kiros et al. [6])	0.5155
Combine-skip (Kiros et al. [6])	0.5209
SIF (Arora et al. [9])	0.4205
Sent2vec Wiki-uni (Pagliardini et al. [7])	0.4279
Sent2vec Wiki-bi (Pagliardini et al. [7])	0.4553
InferSent (Conneau et al. [10])	0.7454
Siamese text embedding (Jeong et al. [20])	0.4263
P-thought (one layer-Bi RNN)	0.7432
P-thought (two layers-Forward RNN)	**0.7899**
P-thought (two layers-Bi RNN)	**0.9725**

similarity scores valued between 0 and 5. The purpose of this task is to approximate the similarity scores between sentences based on the embedded vectors. A description of the dataset is summarized in Table 5.

We conducted the experiment in the same manner as for InferSent. For two sentence vectors u and v, the component-wise product $u \cdot v$ and the absolute difference $|u - v|$ are computed and concatenated to be used as an input. As the target, the human rated similarity score y is transformed as follows. Let $r^T = [1, \ldots, 5]$ denote a vector that takes integer values between 1 and 5. The target y is transformed to the distribution d using the equation below:

Table 4
Extracted sentences for visualization.

Group 1 (△)
1) A woman posing for the camera standing on skis.
2) a woman standing on skiis while posing for the camera.
3) A woman in a red jacket skiing down a slope.
4) A young woman is skiing down the mountain slope.
5) a person on skis makes her way through the snow
Group 2 (◆)
1) A male tennis player in white shorts is playing tennis.
2) This woman has just returned a volley in tennis.
3) A man holding a tennis racket playing tennis.
4) The man balances on one leg after serving a tennis ball.
5) Someone playing in a tennis tournament with a crowd looking on.
Group 3 (■)
1) A woman holding a Hello Kitty phone on her hand.
2) A woman holds up her phone in front of her face.
3) A woman in white shirt holding up a cellphone.
4) A woman checking her cell phone with a hello kitty case.
5) The Asian girl is holding her Miss Kitty phone.
255,215,0Group 4 (×)
1) A plate of food which includes onions, tomato, lettuce, sauce, fries, and a sandwich.
2) A sandwich, french fries, bowl of ketchup, onion slice, lettuce slice, tomato slice, and knife sit on the white plate.
3) Partially eaten hamburger on a plate with fries and condiments.
4) A grilled chicken sandwich sits beside french fries made with real potatoes.
5) A sandwich on a sesame seed bun next to a pile of french fries and a cup of ketchup
Group 5 (▽)
1) Decorated coffee cup and knife sitting on a patterned surface.
2) A large knife is sitting in front of a mug has a skull and crossbones.
3) A white mug showing pirate skull and bones and a large knife on a counter top.
4) There is a white coffee cup with a skull and bones on it next to a knife.
5) A close up of a knife and a cup on a surface

$$d_i = \begin{cases} y - \lfloor y \rfloor, & \text{if } i = \lfloor y \rfloor + 1, \\ \lfloor y \rfloor + 1, & \text{if } i = \lfloor y \rfloor, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Finally, we trained a logistic regression model that predicts the transformed target d from the sentence pair representations of the training dataset. The results for the STS Benchmark test dataset are summarized in Table 6. Fig. 4 presents a scatter plot of the results for the proposed models and the target y.

The experimental results show that the P-thought models of all three levels outperformed the benchmarked models. pecifically, the P-thought model with the lowest complexity exceeds the current best, about 7%. On the other hand, the model with the highest complexity showed only 0.8% of the increase, and the model with the second-highest complexity showed a 5% increase compared to the current best. This opposite tendency for the P-coherence experiment. A more detailed analysis of this phenomenon is discussed in 5.5.

5.4. Semantic relatedness task

We also conducted SemEval 2014 Task 1 which employs semantic relatedness SICK dataset [39]. The dataset consist of 4927 training pairs, 500 development pairs and 4500 test pairs. The purpose of this task is exactly as same as that of STS Benchmark task. Given two sentences and human rated scores, our goal is to produce a similarity score of how semantically these sentences are related. We conducted this task in two different approaches: a supervised approach and an unsupervised approach. The supervised approach was performed in the same way as the STS Benchmark task. We trained a linear model by employing training pairs and measured the Pearson's correlation between similarity score of test pairs and their human rated scores. The unsupervised approach, which was performed in SIF and Sent2vec, was conducted by calculating the Pearson's correlation between cosine similarity of sentence pairs and their human rated scores. The results for the semantic relatedness task are summarized in Table 7. Sup is the result of the supervised task and Uns is the result of the unsupervised task. For the supervised task, our proposed model yielded the second-highest value after that of the InferSent. Similar to the STS Benchmark task, the inverse correlation between the performance and the model complexity is observed. For the unsupervised task, the P-thought model yielded the highest value, outperforming the SIF's result, which is the best performance among benchmarked models. In this case, the model with the highest complexity shows the best result, and the performance

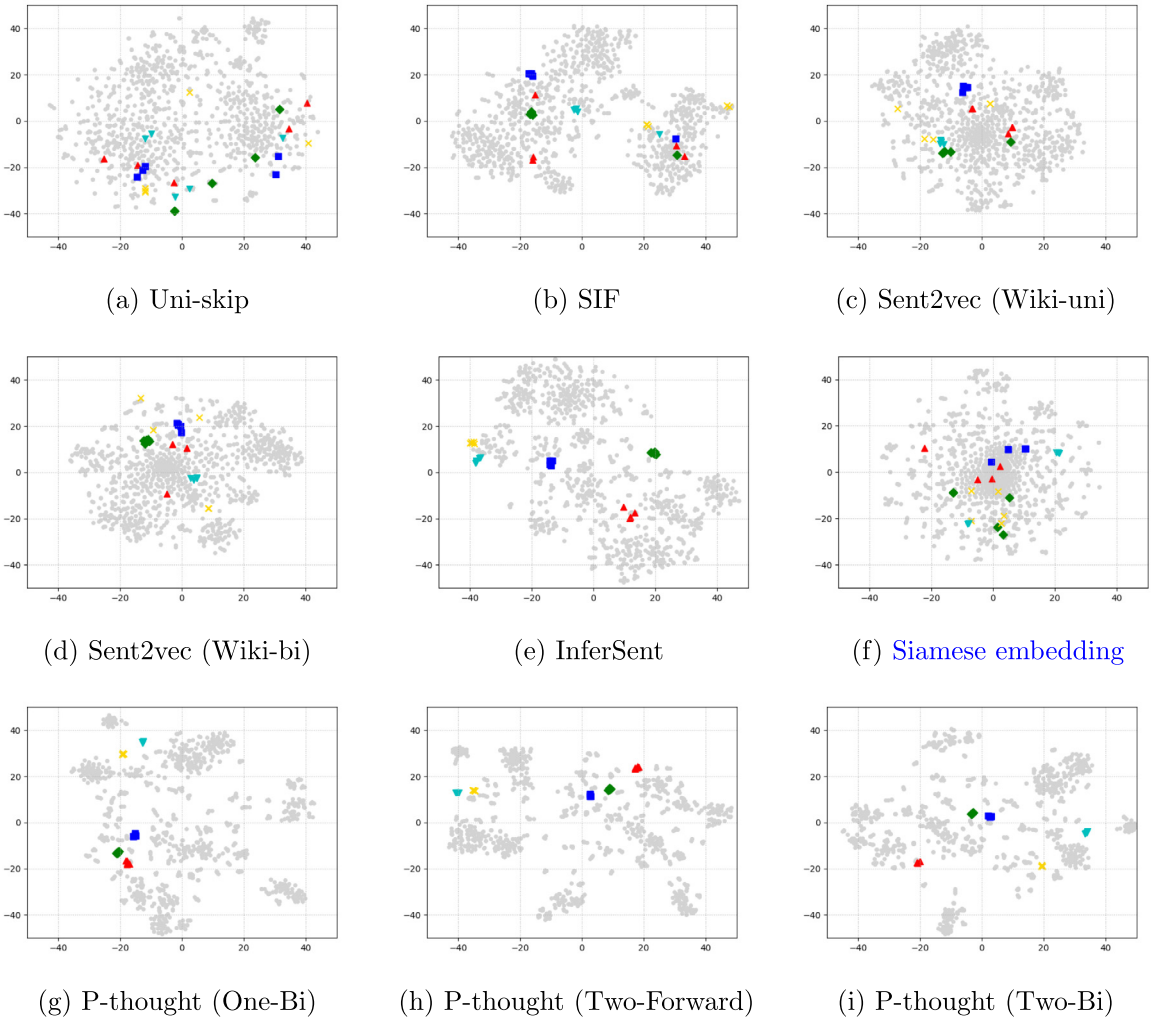


Fig. 3. Scatter plots of the five paraphrase sentence groups represented by each sentence embedding method.

Table 5
STS Benchmark task dataset description.

–	Train	Dev	Test	Total
# of data	5,749	1,500	1,379	8,628

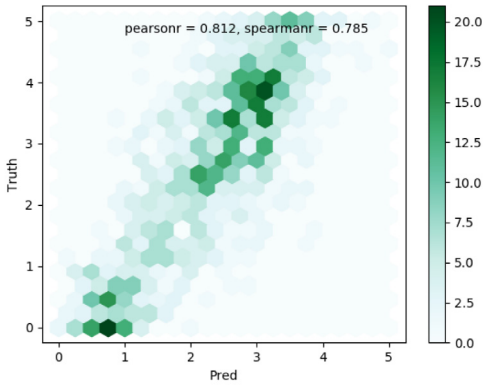
is proportional to the model complexity. However, the increase ratio is not as significant as that of the P-coherence experiment.

5.5. Analysis of the results

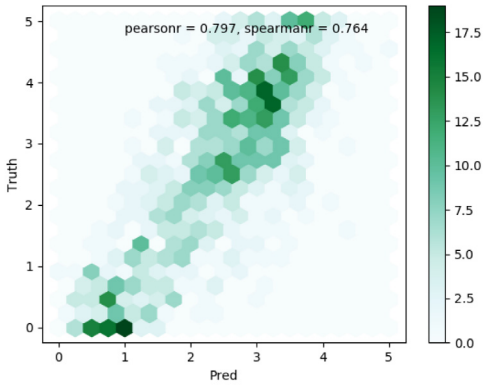
An interesting observation is that while the performances of the P-thought models for the measuring P-coherence task improves as the model complexity increases, the performances of other tasks do not improve as such but slightly degenerates for some cases as shown in Fig. 5. One possible reason for this reversed tendency in performances is that the MS-COCO caption dataset used for the model training consists of only about 600,000 sentences, which are much fewer than training datasets for general sequence learning tasks in the NLP field. Also, the MS-COCO caption dataset lacks variation because redundant phrases occur frequently. Hence, it is more likely to overfit the training dataset with a more complex structure. This problem can be alleviated by obtaining more diversified paraphrase sentence pairs.

Table 6
Experimental results for the STS Benchmark task.

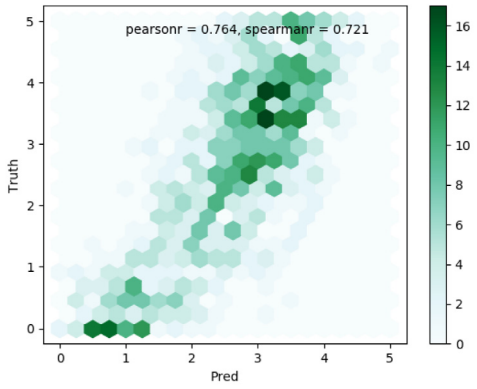
Model	Pearson correlation
PV-DBOW (Le and Mikolov [12], Lau and Baldwin [38])	0.649
SipThought (Kiros et al. [6])	0.721
SIF (Arora et al. [9])	0.720
Sent2vec (Pagliardini et al. [7])	0.755
InferSent (Conneau et al. [10])	0.758
Siamese text embedding (Jeong et al. [20])	0.344
P-thought (one layer-Bi RNN)	**0.812**
P-thought (two layers-Forward RNN)	**0.797**
P-thought (two layers-Bi RNN)	**0.764**



(a) One-layer BiRNN



(b) Two-layer forward RNN



(c) Two-layer Bi-direction RNN

Fig. 4. Correlation scatter plot of STS Benchmark test.

6. Conclusion

Sentence embedding is one of the most important text processing techniques in NLP. To date, various sentence embedding models have been proposed and have yielded good performances in document classification and sentiment analysis tasks. However, the fundamental ability of sentence embedding methods, i.e., how effectively the meanings of the original sentences are preserved in the embedded vectors, cannot be fully evaluated through such indirect methods.

In this study, under the proposition that a good sentence embedding method should act similar to human language recognition, we suggested the concept of semantic coherence and proposed a model named P-thought that aims to maximize the

Table 7
Experimental results for the semantic relatedness task.

Model	Supervised	Unsupervised
PV-DBOW (Le and Mikolov [12], Lau and Baldwin [38])	–	0.460
SkipThought (Kiros et al. [6])	0.858	0.600
SIF (Arora et al. [9])	-	0.719
Sent2vec (Pagliardini et al. [7])	0.860	0.722
InferSent (Conneau et al. [10])	**0.885**	–
Siamese text embedding	0.729	0.071
P-thought (one layer-Bi RNN)	0.861	0.707
P-thought (two layers-Forward RNN)	0.854	0.718
P-thought (two layers-Bi RNN)	0.834	**0.733**

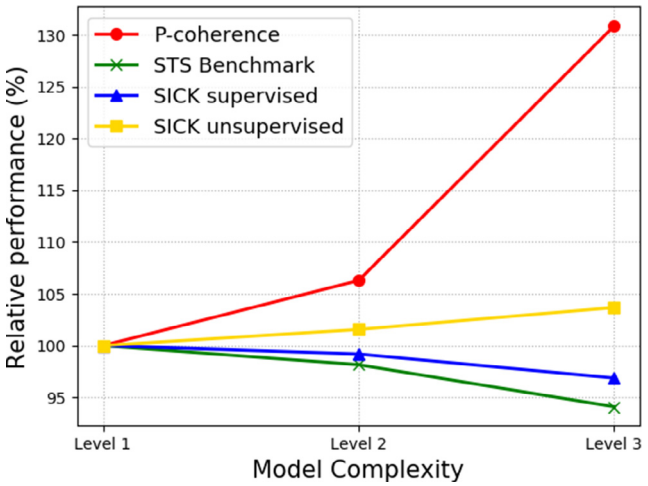


Fig. 5. Relative performances of experimental results.

semantic coherence by designing a model to have a dual generation structure. The proposed model was evaluated based on the MS-COCO caption, STS Benchmark datasets and SICK datasets. Experimental results showed that the P-thought models yielded better performances than the benchmarked models for both tasks. Based on the scatter plots in the two-dimensional space reduced by *t*-SNE, it can clearly be observed that the paraphrase sentences are more concentrated for the P-thought models than those using other sentence embedding methods. The main limitation of the current work is that there are insufficient paraphrase sentences for training the models. P-thought models with more complex encoder structures tend to overfit the MS-COCO datasets. Although this problem can be resolved by acquiring more paraphrase sentences, it is demanding in practice to obtain a large number of paraphrase sentences. Therefore, similar to the approaches that have achieved excellent performances in machine translation by employing semi-supervised learning or unsupervised learning [33,1,40], approaches to improve the performance of the proposed models using only minimal paraphrase data should be developed. Another future research direction is to verify the P-thought using various NLP downstream tasks with different primary objectives rather than semantic similarity used in the current study. If the sentence representation is well-trained, we can expect that this representation will work well for different supervised NLP tasks such as sentiment classification. Hence, it would be worth applying the P-thought embedding to various tasks and comparing its performance with other sentence embedding methods.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Myeongjun Jang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft. **Pilsung Kang:** Conceptualization, Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grants funded by the Korea government (MSIT) (No. NRF-2019R1F1A1060338) and Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0008691, The Competency Development Program for Industry Specialist).

Appendix A. Comparison between P-coherence and normalized Euclidean distance

To evaluate the semantic coherence, we proposed the P-coherence, which computes a summation of cosine similarity between paraphrase sentence vectors. However, high P-coherence does not always ensure semantic adherence when sentence vectors have different norms because the cosine similarity simply measures an angle between two vectors. Although this problem could be alleviated by using the same initialization method, optimizer, and learning rate, the compatibility of the P-coherence should be verified. Therefore, we compared the result of P-coherence with that of normalized Euclidean distance, which is shown in Table A.8. The P-coherence is inversely proportional to the normalized Euclidean distance. Also, we draw a similarity graph for two metrics, as shown in Fig. A.6. To transform the normalized Euclidean distance to a similarity metric, we took a reciprocal of the distance and used a min–max scaling for a fair comparison. The result shows that two similarity metrics show a similar tendency, which approves that vectors with high P-coherence scores are located close to each other.

Table A.8
P-coherence and a normalized Euclidean distance result of P-thought model.

Model complexity	1layer-BiRNN	2layer-forwardRNN	2layer-BiRNN
P-coherence	0.74	0.78	0.97
Normalized Euclidean dist	76.09	75.51	55.03

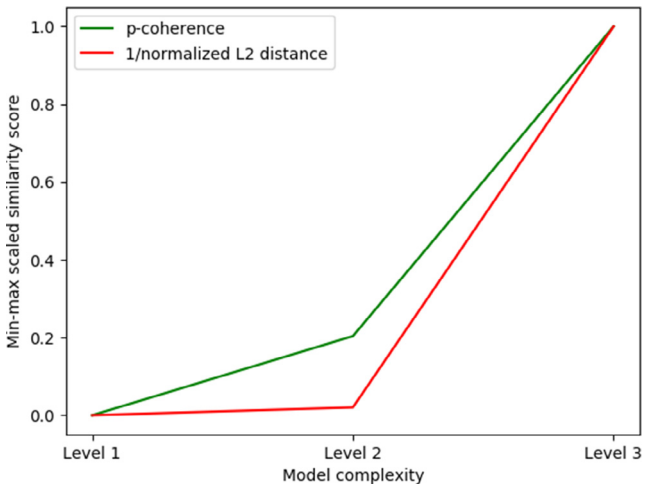


Fig. A.6. P-coherence and an inverse normalized Euclidean distance graph of P-thought model.

Appendix B. Grid search on the value λ

We performed a grid search to find the optimal value of λ which could minimize the objective function. The value of validation loss function according to the value λ is described in Table B.9.

Table B.9
Grid search result for obtaining optimal λ value.

λ	0.005	0.01	0.02	0.05	0.1	0.3	0.5	0.9
loss	1.982	**1.166**	2.012	2.068	2.164	2.543	2.933	3.617

References

[1] M. Artetxe, G. Labaka, E. Agirre, K. Cho, Unsupervised neural machine translation, arXiv preprint arXiv:1710.11041, 2017.
[2] J. Lee, K. Cho, T. Hofmann, Fully character-level neural machine translation without explicit segmentation, arXiv preprint arXiv:1610.03017, 2016.
[3] A. Conneau, H. Schwenk, L. Barrault, Y. Lecun, Very deep convolutional networks for text classification, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, vol. 1, 2017, pp. 1107–1116.
[4] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, B. Xu, Text classification improved by integrating bidirectional lstm with two-dimensional max pooling, arXiv preprint arXiv:1611.06639, 2016.
[5] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, X. Cheng, A deep architecture for semantic matching with multiple positional sentence representations, in: AAAI, vol. 16, 2016, pp. 2835–2841.
[6] R. Kiros, Y. Zhu, R.R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, S. Fidler, Skip-thought vectors, Advances in Neural Information Processing Systems (2015) 3294–3302.
[7] M. Pagliardini, P. Gupta, M. Jaggi, Unsupervised learning of sentence embeddings using compositional n-gram features, arXiv preprint arXiv:1703.02507, 2017.
[8] F. Hill, K. Cho, A. Korhonen, Learning distributed representations of sentences from unlabelled data, arXiv preprint arXiv:1602.03483, 2016.
[9] S. Arora, Y. Liang, T. Ma, A simple but tough-to-beat baseline for sentence embeddings, in: International conference on Learning Representations, 2017.
[10] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data, arXiv preprint arXiv:1705.02364, 2017.
[11] M. Chen, Efficient vector representation for documents through corruption, arXiv preprint arXiv:1707.02377, 2017.
[12] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, 2014, pp. 1188–1196.
[13] M. Iyyer, V. Manjunatha, J. Boyd-Graber, H. Daumé III, Deep unordered composition rivals syntactic methods for text classification, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, 2015, pp. 1681–1691.
[14] K. Soumya George, S. Joseph, Text classification by augmenting bag of words (bow) representation with co-occurrence feature, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278–0661, p-ISSN: 2278-8727, 16 (2014) 34–38.
[15] J. Wieting, M. Bansal, K. Gimpel, K. Livescu, Towards universal paraphrastic sentence embeddings, arXiv preprint arXiv:1511.08198, 2015.
[16] E.H. Huang, R. Socher, C.D. Manning, A.Y. Ng, Improving word representations via global context and multiple word prototypes, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics, 2012, pp. 873–882.
[17] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Advances in Neural Information Processing Systems (2013) 3111–3119.
[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, IEEE, 2009, pp. 248–255.
[19] S.R. Bowman, G. Angeli, C. Potts, C.D. Manning, A large annotated corpus for learning natural language inference, arXiv preprint arXiv:1508.05326, 2015.
[20] M. Jeong, H. Choi, S. Seo, G. Son, K. Park, P. Kang, Measuring sentence similarity based on image association and siamese network, Journal of Korean Institute of Industrial Engineers (2020), 46(2), 123–133.
[21] G. Koch, Siamese neural networks for one-shot image recognition, in: ICML Deep Learning Workshop, 2015.
[22] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, Generative adversarial text to image synthesis, arXiv preprint arXiv:1605.05396, 2016.
[23] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Visionin: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[24] D. Newman, J.H. Lau, K. Grieser, T. Baldwin, Automatic evaluation of topic coherence, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2010, pp. 100–108.
[25] M. Röder, A. Both, A. Hinneburg, Exploring the space of topic coherence measures, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, ACM, 2015, pp. 399–408.
[26] A. Karpathy, A. Joulin, L.F. Fei-Fei, Deep fragment embeddings for bidirectional image sentence mapping, Advances in Neural Information Processing Systems (2014) 1889–1897.
[27] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3128–3137.
[28] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078, 2014.
[29] T. Mikolov, Q.V. Le, I. Sutskever, Exploiting similarities among languages for machine translation, arXiv preprint arXiv:1309.4168, 2013.
[30] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[31] A. Prakash, S.A. Hasan, K. Lee, V. Datla, A. Qadir, J. Liu, O. Farri, Neural paraphrase generation with stacked residual lstm networks, arXiv preprint arXiv:1610.03098, 2016.
[32] A. Gupta, A. Agarwal, P. Singh, P. Rai, A deep generative framework for paraphrase generation, arXiv preprint arXiv:1709.05074, 2017.
[33] Y. Cheng, W. Xu, Z. He, W. He, H. Wu, M. Sun, Y. Liu, Semi-supervised learning for neural machine translation, arXiv preprint arXiv:1606.04596, 2016.
[34] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.
[35] D. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.
[36] L.v.d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (2008) 2579–2605.
[37] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, L. Specia, Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation, arXiv preprint arXiv:1708.00055, 2017.
[38] J.H. Lau, T. Baldwin, An empirical evaluation of doc2vec with practical insights into document embedding generation, arXiv preprint arXiv:1607.05368, 2016.
[39] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, R. Zamparelli, Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 1–8.
[40] G. Lample, L. Denoyer, M. Ranzato, Unsupervised machine translation using monolingual corpora only, arXiv preprint arXiv:1711.00043, 2017.