


Article

Document Re-Ranking Model for Machine-Reading and Comprehension

Youngjin Jang ¹ and Harksoo Kim ^{2,*} ¹ Artificial Intelligence, Konkuk University, Seoul 05029, Korea; dan_yon@konkuk.ac.kr² Computer Science and Engineering & Artificial Intelligence, Konkuk University, Seoul 05029, Korea

* Correspondence: nlpdrkim@konkuk.ac.kr; Tel.: +82-2-450-3499

Received: 26 September 2020; Accepted: 21 October 2020; Published: 27 October 2020

**Featured Application:** Essential technology for practical machine-reading and comprehension systems.

Abstract: Recently, the performance of machine-reading and comprehension (MRC) systems has been significantly enhanced. However, MRC systems require high-performance text retrieval models because text passages containing answer phrases should be prepared in advance. To improve the performance of text retrieval models underlying MRC systems, we propose a re-ranking model, based on artificial neural networks, that is composed of a query encoder, a passage encoder, a phrase modeling layer, an attention layer, and a similarity network. The proposed model learns degrees of associations between queries and text passages through dot products between phrases that constitute questions and passages. In experiments with the MS-MARCO dataset, the proposed model demonstrated higher mean reciprocal ranks (MRRs), 0.8%p–13.2%p, than most of the previous models, except for the models based on BERT (a pre-trained language model). Although the proposed model demonstrated lower MRRs than the BERT-based models, it was approximately 8 times lighter and 3.7 times faster than the BERT-based models.

Keywords: passage re-ranking; passage retrieval; machine-reading comprehension

1. Introduction

Machine-reading and comprehension (MRC) is a question answering task in which computers are required to understand contexts based on passages and answer related questions. With the rapid evolution of deep neural network techniques, the performance of MRC models has been substantially enhanced [1–3]. However, conventional MRC models have deficiencies in that text passages relevant to user queries (i.e., text passages containing phrases answering user queries) should be prepared in advance. Figure 1 illustrates an example in which an MRC model returns different answers according to given passages.

Query: When was Apple established?

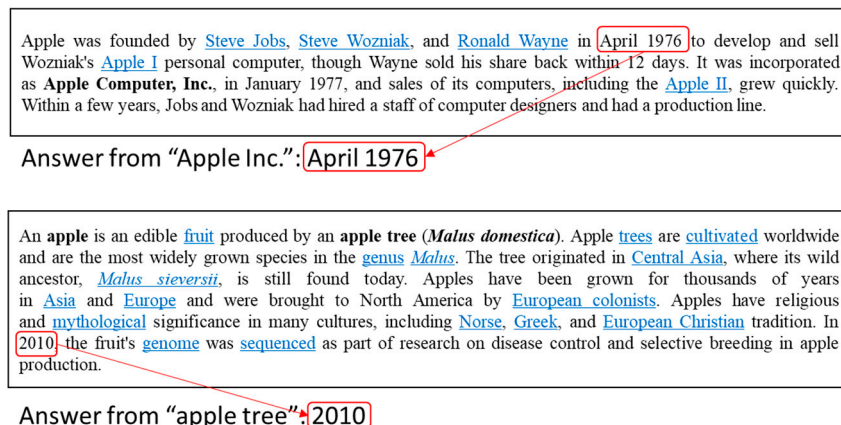


Figure 1. Different answers of a machine-reading and comprehension (MRC) system according to given passages.

To overcome this problem, open-domain MRC models based on information retrieval (IR) have been proposed [4,5]. These models conventionally follow a two-stage process: passage retrieval based on an IR model and answer extraction based on an MRC model, as illustrated in Figure 2.

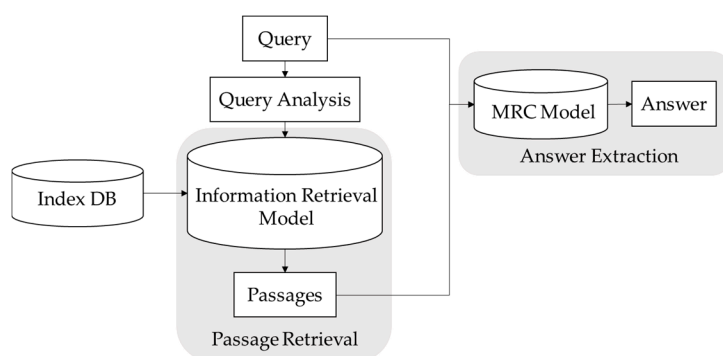


Figure 2. Two-stage process of open-domain MRC.

In several cases, the performance of IR-based MRC models depends on those of underlying IR models that employ term frequency and inversed document frequency (TF-IDF) rankings. As illustrated in Figure 1, when the relevant passage (i.e., the upper document) is given, the MRC model returns the correct answer "April 1975", but when the irrelevant passage (i.e., the lower document) is given, it returns the incorrect answer "2010". Although recent IR models have demonstrated superior performances, the highly ranked documents often do not contain answers to the relevant questions. This leads to a decrease in answer recall in open-domain MRC. Therefore, certain models have been proposed for enhancing IR performance [6]. Similar to Lee et al. [6], we propose an artificial neural network (ANN) model that re-ranks retrieved documents to improve answer recall in MRC (i.e., for ensuring that documents containing answers are ranked high). The proposed model complements the underlying IR model by learning degrees of associations (i.e., possibilities for the documents to contain answer phrases) between queries and documents through a deep neural network.

The remainder of this paper is organized as follows. In Section 2, we briefly review earlier re-ranking models. In Section 3, we describe our model. In Section 4, we explain our experimental setup and report some of our experimental results. In Section 5, we provide the conclusions of our study.

2. Previous Studies

The earlier IR models that employ TF-IDF rankings do not consider semantic information such as homonyms properly because it depends on token-matching methods. To resolve this problem, some IR models based on ANNs have been proposed. Xiong et al. [7] proposed a ranking model called KNRM (Kernel based Neural Ranking Model). It generates a translation matrix that uses similarities between queries and documents. In addition, KNRM used a kernel pooling method to effectively summarize the translation matrix and to generate scores for ranking learning. Guo et al. [8] proposed a ranking model termed DRMM that was based on cosine similarities between query vectors and document vectors in a latent vector space generated by a multilayer perceptron (MLP). DRMM demonstrated superior performance in certain retrieval tasks. However, Dai et al. [9] pointed out that DRMM returns inconsistent similarities based on the lengths of the query and document vectors. To resolve this issue, Dai et al. proposed a cross-mapping function based on a convolutional neural network (CNN) with kernel pooling [7] that always returns fixed lengths of query vectors and document vectors. To overcome the limitation that several ANN models cannot properly reflect term frequency and document frequency, Mitra et al. [10] proposed a joint model in which a local model based on conventional term frequencies and a distributed model based on the distributed representation of words are co-trained. Alaparthi et al. [11] proposed a ranking model that was based on the bi-LSTM with a co-attention mechanism between a query and a document. In addition to co-attention, we also used self-attention mechanism on various word embeddings (e.g., word2vec [12], GloVe [13], fastText [14]).

3. Re-Ranking Model Based on Artificial Neural Network

Figure 3 illustrates the overall architecture of the proposed re-ranking model. As depicted, the proposed model consists of five parts: query encoder, passage encoder, phrase modeling layer, attention layer, and similarity network. In this paper, the term “passage” refers to an indexing unit that is typically referred to as a document in IR.

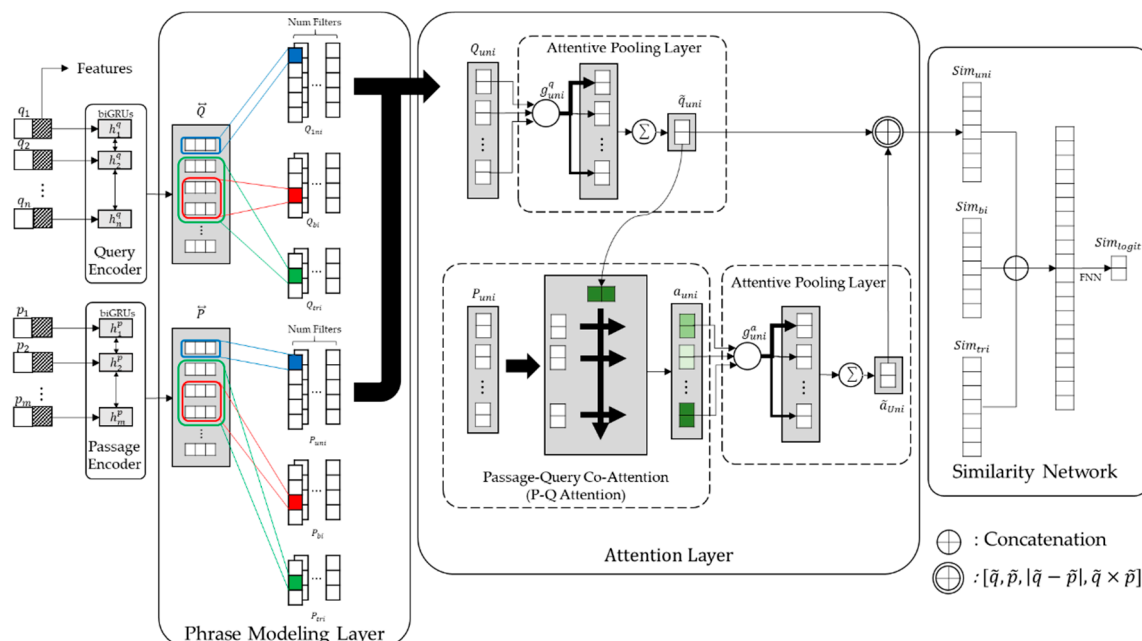


Figure 3. Overall architecture of proposed model.

The input units of the query and passage encoders are words, and each word is represented by a concatenation of four types of embeddings, as depicted in Figure 4.

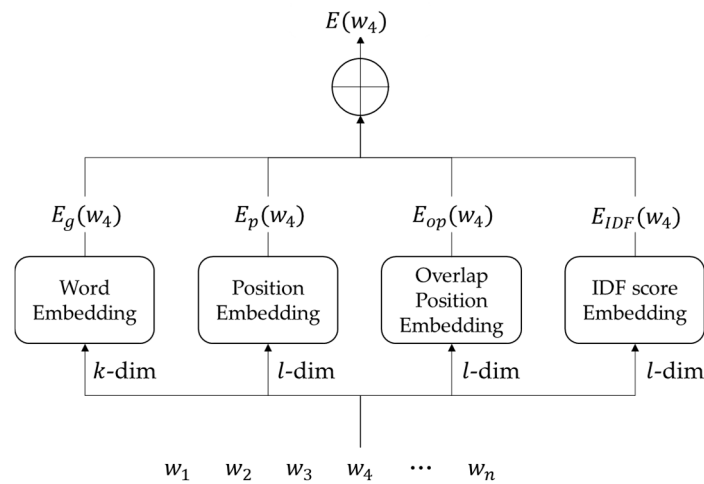


Figure 4. Input unit of proposed model.

In Figure 4, w_i is the i th word in a query or a passage and $E_g(w_i)$ is a pre-trained k -dimensional GloVe embedding [13] of w_i . Thus, $E_p(w_i)$ is an l -dimensional position embedding of w_i that represents a word position in the query or passage, and $E_{IDF}(w_i)$ is an l -dimensional inversed document frequency (IDF) embedding [15] that is set to a discrete value according to the score intervals of 0.05. Further, $E_{op}(w_i)$ is an l -dimensional position embedding of an overlapped word in the other text between the query and the passage. For example, when the query “I go to school” and the passage “We should come back to school” are presented, $E_{op}(w_4)$ of the overlapping word “school” in the query is set to 6, meaning that the overlapping word occurs in the 6th position in the passage (opponent text). We empirically set the embedding sizes of $E_p(w_i)$, $E_{op}(w_i)$, and $E_{IDF}(w_i)$ to all the same dimensions because we cannot distinguish them in terms of the quantity of information. All embeddings except the GloVe embedding are randomly initialized and fine-tuned during training. To simplify the equations, we rewrite $E(w_i)$ for the i th input unit in a query and a passage as q_i and p_i , respectively.

The query and passage encoders convert the query vector $Q = (q_1, q_2, \dots, q_n)$ with n word vectors and the passage vector $P = (p_1, p_2, \dots, p_m)$ with m word vectors into the encoded vectors, \vec{Q} and \vec{P} , respectively, which embed contextual information using bidirectional gated recurrent units (biGRUs) [16], as represented by Equation (1):

$$\begin{aligned}
 \vec{h}_i^Q &= \text{GRU}\left(E(q_i), \vec{h}_{i-1}^Q\right), \overleftarrow{h}_i^Q = \text{GRU}\left(E(q_i), \overleftarrow{h}_{i+1}^Q\right), \overleftrightarrow{h}_i^Q = \begin{bmatrix} \vec{h}_i^Q & \overleftarrow{h}_i^Q \\ \overrightarrow{h}_i^P & \overleftarrow{h}_i^P \end{bmatrix} \\
 \overrightarrow{h}_i^P &= \text{GRU}\left(E(p_i), \overrightarrow{h}_{i-1}^P\right), \overleftarrow{h}_i^P = \text{GRU}\left(E(p_i), \overleftarrow{h}_{i+1}^P\right), \overleftrightarrow{h}_i^P = \begin{bmatrix} \overrightarrow{h}_i^P & \overleftarrow{h}_i^P \\ \vec{h}_i^Q & \overleftarrow{h}_i^Q \end{bmatrix} \\
 \overleftrightarrow{Q} &= \left\{ \overleftrightarrow{h}_1^Q, \overleftrightarrow{h}_2^Q, \dots, \overleftrightarrow{h}_n^Q \right\}, \overleftrightarrow{P} = \left\{ \overleftrightarrow{h}_1^P, \overleftrightarrow{h}_2^P, \dots, \overleftrightarrow{h}_m^P \right\}
 \end{aligned} \tag{1}$$

In Equation (1), $\left[\vec{h}_i, \overleftarrow{h}_i\right]$ is the concatenation of a forward hidden state \vec{h}_i and a backward hidden state \overleftarrow{h}_i . The weights in the query encoder and passage encoder are not shared. Thus, the encoded-word vectors (outputs of the query encoder and the passage encoder) are input to the phrase-modeling layer.

The phrase-modeling layer generates phrase-level features based on word n -grams (from word unigram to word trigram) using CNNs [17]. The CNNs used in the phrase modeling layer do not have any pooling layers, unlike conventional CNNs, as depicted in Figure 5.

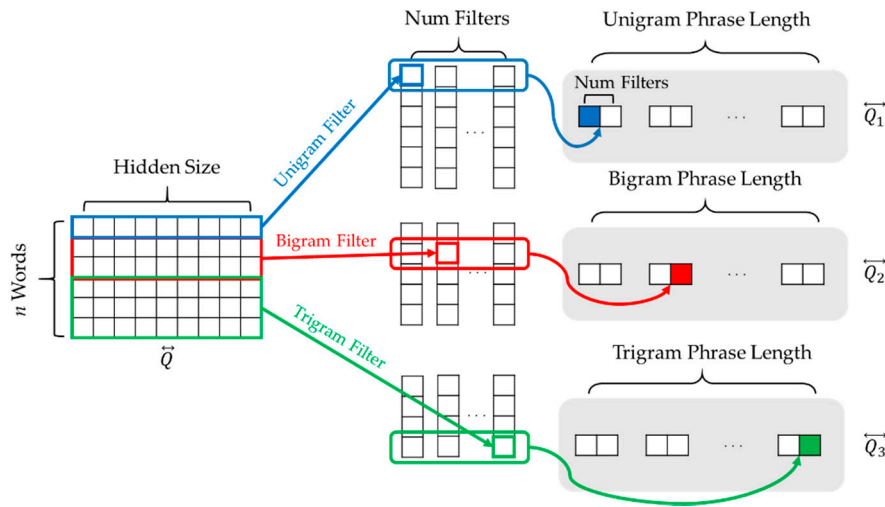


Figure 5. Generation of phrase-level features using convolutional neural network (CNN)s.

Using the CNNs depicted in Figure 5, the encoded vectors \vec{Q} and \vec{P} are represented as three types of phrase vectors: unigram phrase vectors \vec{Q}_1 and \vec{P}_1 , bigram phrase vectors \vec{Q}_2 and \vec{P}_2 , and trigram phrase vectors \vec{Q}_3 and \vec{P}_3 . To simplify the equations, we rewrite the n -gram phrase vectors as \vec{Q}_n and \vec{P}_n . Thereafter, the n -gram phrase vectors are input to the attention layer.

The attention layer consists of two sublayers: a passage–query (P–Q) attention layer and an attention pooling layer. In the P–Q attention layer, the proposed model calculates the degrees of associations between the n -gram phrases in a query and those in a passage. The P–Q attention vector of an n -gram phrase vector, P_{att}^n , is calculated using the scaled dot product [18] expressed in Equation (2):

$$P_{att}^n = \text{softmax} \left(\frac{\vec{P}_n \cdot \vec{Q}_n^T}{\sqrt{d_{\vec{Q}_n}}} \right) \vec{P}_n \quad (2)$$

In Equation (2), \vec{P}_n denotes an n -gram phrase vector of a passage, and \vec{Q}_n denotes an n -gram phrase vector of a query mapped onto the $d_{\vec{Q}_n}$ dimension using a conventional pooling mechanism [19]. That is, the proposed model converts \vec{Q}_n and P_{att}^n into fixed-length vectors in the attentive pooling layer, as expressed in Equation (3):

$$\begin{aligned} g_n^q &= \text{softmax} \left(w_n^q \left(\vec{Q}_n \right) + b_q \right) \\ \tilde{Q}_n &= \left(\sum g_n^q \times \vec{Q}_n \right) \\ g_n^p &= \text{softmax} \left(w_n^p \left(P_{att}^n \right) + b_p \right) \\ \tilde{P}_n &= \left(\sum g_n^p \times P_{att}^n \right) \end{aligned} \quad (3)$$

In Equation (3), w_n and b_n denote a weight matrix and bias vector, respectively. Therefore, g_n^q and g_n^p are generated by feed-forward neural networks (FNNs). Then, \times denotes a cross product between two vectors. The normalized attention vectors \tilde{Q}_n and \tilde{P}_n for the n -gram phrase vectors are input to the similarity network.

To calculate the similarity between the normalized attention vector \tilde{P}_n and the normalized query vector \tilde{Q}_n , we adopt the similarity vector representation proposed in a report on sentence embedding by Conneau et al. [20], as expressed in Equation (4):

$$\text{sim}(\tilde{Q}_n, \tilde{P}_n) = \left[\tilde{Q}_n, \tilde{P}_n, \left| \tilde{Q}_n - \tilde{P}_n \right|, \tilde{Q}_n \times \tilde{P}_n \right] \quad (4)$$

In Equation (4), $[\tilde{Q}, \tilde{P}, \dots]$ and \times denote the concatenation of vectors and a cross product, respectively. The final logit function for the similarity calculation between a query and a passage is expressed in Equation (5):

$$\text{sim}_{\text{logit}}(\tilde{Q}, \tilde{P}) = w \cdot \left[\text{sim}(\tilde{Q}_1, \tilde{P}_1), \text{sim}(\tilde{Q}_2, \tilde{P}_2), \text{sim}(\tilde{Q}_3, \tilde{P}_3) \right] + b. \quad (5)$$

In Equation (5), $[\text{sim}(\tilde{Q}_1, \tilde{P}_1), \dots]$ denotes the concatenation of n -gram similarity vectors. Thus, w and b denote a weight matrix and bias vector, respectively, to represent similarity distributions through an FNN. Finally, the model is trained using cross-entropy loss, as expressed in Equation (6):

$$\text{loss} = - \sum_i y_i \log(\text{softmax}(\text{sim}_{\text{logit}}^i)) \quad (6)$$

4. Evaluation

4.1. Datasets and Experimental Settings

We trained and evaluated our model on the Microsoft Machine-Reading Comprehension (MS-MARCO) dataset [21]. The training set contains approximately 400 M tuples of queries and relevant and non-relevant passages. The development set contains approximately 6900 queries, each paired with the top 1000 passages retrieved with BM25 [22] from the MS-MARCO dataset. On average, each query has one relevant passage. We trained the model via negative sampling using the ratio of 1:5 for the 1000 passages corresponding to each query. To implement the proposed model, we adopted the pre-trained GloVe algorithm. The vocabulary size of GloVe was 300. We set the hidden size of the GRU neural network to 200. The model optimization was performed with Adam [23] at a learning rate of 0.0001, and the learning rate was halved if the validation performance did not improve. The dropout rate was set to 0.2, and the mini-batch size was set to 256 sequences.

We used mean reciprocal rank at 10 (MRR@10) [24], which represents the MRR score of documents ranked in the top 10 because it is essential for relevant documents to be highly ranked for MRC models, as expressed in Equation (7):

$$\text{MRR@10} = \frac{1}{n} \sum_{i=1}^{10} \frac{1}{r_i} \quad (7)$$

In Equation (7), r_i is the rank of the first passage containing a correct answer produced by the i th query and n is the number of queries.

4.2. Experimental Results

The first experiment was conducted to evaluate the effectiveness of the additional input embeddings (i.e., position embedding, overlapped position embedding, and IDF embedding) and the phrase modeling layer by comparing the changes in performance, as presented in Table 1.

Table 1. Performance changes in development set.

Model	MRR@10
Proposed model	0.303
w/o additional input embeddings	0.223
w/o phrase modeling layer	0.295

In Table 1, “w/o additional input embeddings” refers to a modified model in which only word embeddings are used as input units. Therefore, “w/o phrase modeling layer” means a modification of our model in which the phrase modeling layer is excluded. As presented in Table 1, the additional input embeddings and phrase modeling layer contribute to the improvement of MRR@10 by 8%p and 0.8%p, respectively. In addition, to check whether the word n -gram features can effectively contain phrase information or not, we visualized the degrees of associations between the n -gram phrases in a query and those in a passage (i.e., P–Q attention scores in Equation (2)) through 2-dimensional heat maps, as shown in Figure 6.

	Uni-gram example	Bi-gram example	Tri-gram example
Given Query		what does job costs include	
Heat map	job costing is the process of assigning costs to custom products or services direct materials and direct labor are traced to individual jobs and production overhead is allocated manufacturers that use job costing include aircraft builders custom motorcycle and auto mobile manufacturers and custom designed jewelers among others	job costing is the process of assigning costs to custom products or services direct materials and direct labor are traced to individual jobs and production overhead is allocated manufacturers that use job costing include aircraft builders custom motorcycle and auto mobile manufacturers and custom designed jewelers among others	job costing is the process of assigning costs to custom products or services direct materials and direct labor are traced to individual jobs and production overhead is allocated manufacturers that use job costing include aircraft builders custom motorcycle and auto mobile manufacturers and custom designed jewelers among others
Given Query		what does dol neg grant stand for	
Heat map	adjustment assistance taa national emergency grants neg and special response grants srr design framework a set of activities that give a local youth program structure and establish coordination among case managers and service providers charged with serving youth	adjustment assistance taa national emergency grants neg and special response grants srr design framework a set of activities that give a local youth program structure and establish coordination among case managers and service providers charged with serving youth	adjustment assistance taa national emergency grants neg and special response grants srr design framework a set of activities that give a local youth program structure and establish coordination among case managers and service providers charged with serving youth

Figure 6. Heat map for visualizing the degrees of associations between a query and a passage.

In Figure 6, the n -gram phrases with higher attention scores were colored in bluer. As illustrated in Figure 6, the uni-gram features were colored in bluer between single words or short phrases, and the tri-gram features were colored in bluer between long phrases. It reveals that each n -gram feature differently contributes to capturing associations between a query and a passage.

The second experiment was conducted to compare the proposed model with the earlier models, as presented in Table 2.

Table 2. Performance comparison.

Model	MRR@10	
	Development Set	Test Set
BM25 [22]	0.167	0.167
KNRM [7]	0.218	0.218
Duet v2 [10] (official baseline)	0.243	0.245
Duet v2 [10] (ensembled)	0.252	0.252
Conv-KNRM [9]	0.247	0.247
Conv-KNRM [9] (ensembled)	0.271	0.290
Alaparthi et al., 2019 [11]	0.298	0.291
Proposed model	0.303	0.299
BERT-Base [25]	0.347	0.347
BERT-Large [25]	0.365	0.365

Referring to Table 2, BM25 is a traditional retrieval model termed Okapi BM25, and Duet v2 is a joint ANN model comprising a local model based on term frequencies and a distributed

model based on word vectors. Conv-KNRM is an ANN model in which queries and documents are encoded using CNNs. Alaparthi et al. [11] refer to a bidirectional long short-term memory network model with a co-attention mechanism between query and passage representations. BERT-Base and BERT-Large are fine-tuned classification models based on the base and large models of BERT [25], which is a pre-trained language model with state-of-the-art performance in several downstream natural language processing tasks such as span prediction, sequence labeling, and text classification [26]. As presented in Table 2, the proposed model outperformed all the previous models except the BERT-based models (The proposed model named “n-gram co-attention” can be found in the official rankings: <https://microsoft.github.io/msmarco/>). Table 3 presents the memory usages and the response times of the proposed and BERT-based models.

Table 3. Comparison of memory usage.

Model	Memory Usage (MB)	Training Parameters (MB)	Response Time (ms)
BERT-Base [25]	1289	110	1100
Proposed model	161	3.5	300

In Table 3, the response time is the average time per query that is spent to rank the top 1000 passages retrieved with BM25 [22]. In order to re-rank the passages retrieved against 6900 queries in the development set, BERT-Base spent approximately 2.108 h, but the proposed model spent approximately 0.575 h. As presented in Table 3, although the proposed model demonstrated lower performance than the BERT models, it demonstrated significantly less memory usage (about 8.0 times less) and faster response time (about 3.7 times faster) than the latter. The ratio between the response times, $300/1100 \approx 0.27$, is bigger than the ratio between the sizes of parameters, $3.5/110 \approx 0.03$. It is caused by the difference of neural network frameworks: the recurrent neural network framework used for the proposed model should sequentially process input words, but the transformer framework used for BERT-Base can process all input words in parallel. Based on these experimental results, we conclude that the proposed model may be more suitable for practical open-domain MRC systems that should respond to multiple substantial user queries simultaneously.

5. Conclusions

We proposed an ANN-based model to re-rank documents retrieved using a conventional IR model, BM25, to improve the performance of MRC models. The proposed model was composed of five subnetworks: query encoder, passage encoder, phrase modeling layer, attention layer, and similarity network. By calculating the mutual information of the phrase unit for queries and passages, the passage scores for queries were effectively reflected. In the experiments with the MS-MARCO dataset, the proposed model demonstrated better MRRs, 0.8%p–13.2%p, than the previous models, except for the BERT-based models. Although the proposed model demonstrated lower MRRs than the BERT-based models, it demonstrated significantly more efficient (approximately 8 times less memory usage and approximately 3.7 times faster response time) than the latter in terms of memory usage and response time. We conclude that these efficiencies are very important engineering factors in the development of a practical MRC system for massive concurrent users.

Author Contributions: Conceptualization and methodology, H.K.; software, validation, formal analysis, investigation, resources, data curation, and writing—original draft preparation, Y.J.; writing—review and editing, visualization, supervision, project administration, and funding acquisition, H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by Konkuk University in 2020.

Acknowledgments: We are grateful for the technical support of the members of the NLP laboratory in Konkuk University.

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

References

1. Lee, H.-G.; Kim, H. GF-Net: Improving machine reading comprehension with feature gates. *Pattern Recognit. Lett.* **2020**, *129*, 8–15. [[CrossRef](#)]
2. Wang, W.; Yang, N.; Wei, F.; Chang, B.; Zhou, M. Gated Self-Matching Networks for Reading Comprehension and Question Answering. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, 30 July–4 August 2017; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA; pp. 189–198.
3. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In Proceedings of the 8th International Conference on Learning Representations 2020 (ICLR), Addis Ababa, Ethiopia, 27–30 April 2020.
4. Chen, D.; Fisch, A.; Weston, J.; Bordes, A. Reading Wikipedia to Answer Open-Domain Questions. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, 30 July–4 August 2017; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA; pp. 1870–1879.
5. Lee, J.; Yun, S.; Kim, H.; Ko, M.; Kang, J. Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA; pp. 565–569.
6. Kratzwald, B.; Feuerriegel, S. Adaptive Document Retrieval for Deep Question Answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA; pp. 576–587.
7. Xiong, C.; Dai, Z.; Callan, J.; Liu, Z.; Power, R. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, 7–11 August 2017; Association for Computing Machinery (ACM): New York, NY, USA; pp. 55–64.
8. Guo, J.; Fan, Y.; Ai, Q.; Croft, W.B. A Deep Relevance Matching Model for Ad-hoc Retrieval. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; Association for Computing Machinery (ACM): New York, NY, USA; pp. 55–64.
9. Dai, Z.; Xiong, C.; Callan, J.; Liu, Z. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining—WSDM, Marina Del Rey, CA, USA, 5–9 February 2018; Association for Computing Machinery (ACM): New York, NY, USA; pp. 126–134.
10. Mitra, B.; Craswell, N. An Updated Duet Model for Passage Re-Ranking. *arXiv* **2019**, arXiv:1903.07666v1.
11. Alaparthi, C. Microsoft AI Challenge India 2018: Learning to Rank Passages for Web Question Answering with Deep Attention Networks. In Proceedings of the 2nd Workshop on Humanizing AI(HAI) at IJCAI'19, Macao, China, 12 August 2019.
12. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
13. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA; pp. 1532–1543.
14. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [[CrossRef](#)]
15. Cover, T.M. *Elements of Information Theory*; John Wiley & Sons: Hoboken, NJ, USA, 1999.
16. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In Proceedings of the NIPS 2014 Workshop on Deep Learning NeurIPS, Montreal, QC, Canada, 8–13 December 2014.
17. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A Convolutional Neural Network for Modelling Sentences. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 23–25 June 2014; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA; pp. 655–665.

18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, L. Attention Is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–7 December 2017; pp. 5998–6008.
19. Zhou, X.; Wan, X.; Xiao, J. Attention-based LSTM Network for Cross-Lingual Sentiment Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA; pp. 247–256.
20. Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; Bordes, A. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA; pp. 670–680.
21. Bajaj, P.; Campos, D.; Craswell, N.; Deng, L.; Gao, J.; Liu, X.; Majumder, R.; McNamara, A.; Mitra, B.; Nguyen, T.; et al. MS MARCO: A Human Generated MACHine Reading COMprehension Dataset. *Choice* **2016**, *2640*, 660.
22. Yang, P.; Fang, H.; Lin, J. Anserini: Reproducible ranking baselines using lucene. *ACM J. Data Inf. Qual.* **2018**, *10*, 16. [[CrossRef](#)]
23. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 7–9 May 2015.
24. Craswell, N. Mean Reciprocal Rank. In *Encyclopedia of Database Systems*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2009; p. 1703.
25. Nogueira, R.; Cho, K. Passage Re-Ranking with BERT. *arXiv* **2019**, arXiv:1901.04085.
26. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; (Long and Short Papers). Volume 1, pp. 4171–4186.

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



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).