A Near-Real-Time Answer Discovery for Open-Domain With Unanswerable Questions From the Web

MINTAE KIM, SANGHEON LEE, YEONGTAEK OH, HYUNSEUNG CHOI, AND WOOJU KIM

Department of Industrial Engineering, Yonsei University, Seoul 03722, South Korea

Corresponding author: Wooju Kim (wkim@yonsei.ac.kr)

This work was financially supported by Korea Ministry of Land, Infrastructure and Transport (MOLIT) as [Innovative Talent Education Program for Smart City]. This work was also supported by SK Telecom Company, Ltd.

ABSTRACT With the proliferation of question and answering (Q&A) services, studies on building a knowledge base (KB) using various information extraction (IE) methodologies from unstructured data on the Web have received significant attention. Existing IE approaches, including machine reading comprehension (MRC), can find the correct answer to a question if the correct answer exists in the document. However, most are prone to extracting incorrect answers rather than producing no answers when the correct answer does not exist in the given documents. This problem is likely to cause serious real-world problems when we apply such technologies to practical services such as AI speakers. We propose a novel open-domain IE system to alleviate the weaknesses of previous approaches. The proposed system integrates an elaborated document selection, sentence selection, and knowledge extraction ensemble method to obtain high specificity while maintaining a realistically achievable level of precision. Based on this framework, we extract answers on Korean open-domain user queries from unstructured documents collected from multiple Web sources. For evaluating our system, we build a benchmark dataset with the SKTelecom AI Speaker log. The baseline models KYLIN infobox generator and BiDAF were used to evaluate the performance of the proposed approach. The experimental results demonstrate that the proposed method outperforms the baseline models and is practically applicable to real-world services.

INDEX TERMS Information extraction system, knowledge base, natural language processing, neural networks, real time systems.

I. INTRODUCTION Formal knowledge bases (KBs), such as the Linked Open Data Cloud (LOD) [1] are used to express and share knowledge by connecting and assigning resources on the Web. The KB is a core element used in question and answering (Q&A) service systems and is considered an important research subject in the field of artificial intelligence as a technology storing and searching for answers to a user query.

Previous studies on information extraction (IE) can be classified into three types. The first type requires creating an IE rule by an expert in a specific domain and extract the knowledge when a matching rule pattern is found in the document. Rule-based IE usually exhibits high performance only in specific documents because knowledge is extracted only for specific types of patterns. Consequently, the cost of using domain experts is high, with the burden of continually adding new patterns. The second type requires extracting information based on supervised machine learning and deep learning models. In this model-based IE, the information is only sufficiently extracted when data have the same structure as the training data. However, for data in a different form, developing capable IE is challenging. The third type requires the study of machine reading comprehension (MRC). In this case, the information is extracted under the assumption that there is a correct answer in the document, such as in the Stanford Question Answering Dataset (SQuAD) [2]. This MRC might result in poor performance on unstructured documents on the Web because it cannot guarantee that the retrieved document contains correct answers.

IE for KB extension should be capable of dealing with diverse types of documents collected from multiple sources...
existing on the Web. Therefore, we require a method to decide which source is more reliable than others. Furthermore, we require a measure to judge whether each retrieved document and each sentence in the document contains correct answers for the subject.

In this study, we propose a novel IE system that can respond practically to open-domain queries, including unanswerable questions. The proposed method consists of a suitable document collection step, a sentence classification step, a knowledge extraction step, a post-processing step, and a final ensemble step. We empirically confirmed that our proposed method could extract highly-reliable knowledge. The extracted knowledge is converted into triple form and stored in the KB. The KB constructed in this way can be used for artificial intelligence (AI)-based technology in the future. KYLIN infobox generator [3] and BiDAF [4] MRC models were selected as baselines to verify performance.

II. RELATED WORK
A. INFORMATION EXTRACTION
IE is a technique for automatically extracting information from a large number of structured or unstructured documents for a given user query [5]. For example, given a user query (“Leaning Tower of Pisa,” “Height”), you can extract the triple (“Leaning Tower of Pisa,” “Height,” “55.86m”) from the sentence “Leaning Tower of Pisa, designed by Italian genius architect Bonanno Pisano, is a bell tower of 55.86m high and 16m diameter” in the “Leaning Tower of Pisa” Wikipedia page. The extracted information about the user query may be stored in a KB to be used in the question answering system or may be directly provided to the user as an answer. IE can be classified according to the IE methodology and document type [6].

The two IE methodologies are (1) knowledge engineering and (2) automatic training. The knowledge engineering methodology is a grammatical rule based on domain knowledge, which defines a pattern of IE and extracts information when a sentence is found that matches the pattern. The automatic training methodology generates label data to train a model and extract information into the trained model. The knowledge engineering methodology is suitable for extracting information on known patterns. However, extracting new types of information is challenging and requires the efforts of domain experts [7]. Therefore, the automatic training methodology has been the most extensively studied [8].

The three IE document types are (1) unstructured text, (2) structured text, and (3) semi-structured text. Unstructured documents, which are various documents on the Web, are the primary targets in the field of IE. Information is extracted through natural language processing techniques and rule-based systems. Structured documents are those that have a predefined structured format, and information is extracted through a relatively simple technique. Semi-structured documents are documents that do not have a fixed format (e.g., HTML or tables) and extract information through patterns such as tokens and separators that are appropriate for each situation. Because the Web is composed predominantly of texts, IE research has been used in various ways as the chief technology for discovering knowledge on the Web [6].

In previous studies on IE methodology, Etzioni et al. [9] proposed a Web-scale domain-independent IE methodology KNOWITALL that used an ontology KB and rule templates to create IE rules for ontology classes and relationships; it measures the reliability of extraction results based on the Naïve Bayes classifier. Banko et al. [8] developed TEXTRUNNER to extract reliable relationship triples from Web documents after building self-supervised data through dependency parsing. Wu et al. [3] proposed an IE system KYLIN for generating infobox from Wikipedia documents by constructing a training dataset by mapping the values of Wikipedia documents and infobox. KNOWITALL, TEXTRUNNER extracts information by finding a pattern that meets a predefined rule but is limited by the challenge of applying IE to data having a new pattern that does not meet a rule. Moreover, TEXTRUNNER designed a self-supervised learning model using Wikipedia and developed a system for extracting information, assuming Wikipedia document types. Consequently, TEXTRUNNER illustrates low performance on heterogeneous data that differs from training data. KYLIN extracts information by selecting a specific model, from several models that exist, based on category-attribute. Therefore, if the document category is misclassified or no model corresponds to the classified category-attribute, it is impossible to extract the information.

B. MACHINE READING COMPREHENSION
MRC is a task used to test how accurately a machine can understand natural language by asking the machine to answer questions based on a given context [10]. MRC research based on deep learning (i.e., neural MRC) has attracted recent attention.

Many studies [4], [11], [12] have reported positive results with recurrent neural networks (RNNs). Herman et al. [11] proposed a “document-query-answer” triple generation method using the RNN with attention for the CNN/DailyMail dataset. Wang et al. [12] proposed an MRC model that matches the document with the query and reflects the attention weight in the query. Seo et al. [4] proposed the BiDAF model for improved performance through the bidirectional-attention-based matching of context and queries. Furthermore, several studies [13], [14] use self-attention [15] structures to efficiently reflect context information and reduce computation. Devlin et al. [13] proposed Bidirectional Encoder Representations from Transformers (BERT) using a transformer [15] structure composed of convolution and self-attention. Yu et al. [14] proposed Q&A architecture that reflects local interaction and global interaction using self-attention.

However, previous studies primarily target cases in which the correct answer always exists in the document, such as SQuAD, NewsQA [16], and MCTest [17]. Consequently,
no procedure exists for judging whether the correct answer is included in the document. Furthermore, applying MRC studies to unstructured documents on the Web is inadequate because the attempts to extract information frequently occur even in documents that do not have correct answers.

III. METHODOLOGY

The proposed methodology is a system to extract answers from Korean user queries based on subject-predicate (SP) from the unstructured documents collected from multiple Web sources. Figures 1 and 2 depict the inference example and architecture of our IE system, which consists of five steps: seed and train data generation, document selection, sentence selection, IE, and knowledge ensemble. The training data generation step generates data for training the model of each module. Furthermore, all data used to train the model in this study were generated using Wikipedia. The document selection module collects relevant documents from Wikipedia, Naver Encyclopedia, and Naver News Web sources for a given Korean user query and determines whether the collected documents are suitable for extracting information. The sentence selection module separates the document into sentences. It selects a sentence, including answer information about a user query, using three methods: sentence matching rules, predicate-based support vector machine (SVM), and sentence-based convolutional neural network (CNN). The IE module extracts the answer from the sentence selected in the sentence selection module. Next, the information extracted from the above extractor is normalized using post-processing. Finally, in the ensemble module, the results of each model are integrated to extract final results and confidence scores.

A. SEED AND TRAIN DATA GENERATION

We generated seed data using Korean Wikipedia to create training data for each step model. As depicted in Figure 3, a Wikipedia page contains the main text and an infobox that summarizes the information of the page. The seed data was generated by extracting Title, Attribute, Sentence, Sentence Label, and Value by mapping the text and infobox value of the Wikipedia page. For example, on the “Leaning Tower of Pisa” Wikipedia page in Figure 3, the height attribute value of the infobox is 55.86 meters. Accordingly, the label of the sentence containing the height attribute value was set to 1, and the sentence without the attribute value was set to 0. Table 1 illustrates an example of seed data.

Table 1

| Model      | Source | Result | Score |
|------------|--------|--------|-------|
| IEM-PB(P)  | Wiki   | 55.86m | 0.78  |
| IEM-PB(A)  | Ency   | 55.86m | 0.47  |
| IEM-SPB(A) | Wiki   | 55.86m | 0.28  |
| IEM-SPB(A) | News   | 64m    | 0.66  |

Next, the information extracted from the above extractor is normalized using post-processing. Finally, in the ensemble module, the results of each model are integrated to extract final results and confidence scores.
Moreover, we trained the IE model using Title, Attribute, Sentence, and Value columns in the seed data. Accordingly, after extracting the columns, we tokenized the sentence and tagged the sections of value. The data collected for training the IE model included approximately 1 million sentences.

### B. DOCUMENT SELECTION MODULE

In the document selection module, we create a search keyword with an SP Korean user query to search and collect documents. A search keyword was created for the SP Korean query log. Based on the created search keywords, the relevant documents were collected from Wikipedia, Naver Encyclopedia, and Naver News Web sources, and then documents were selected using the document selection rule. Table 2 summarizes the search keywords, document collection methods, and proper document selection rules for each of the three sources.

#### TABLE 2. Document collection and selection methods.

| Source       | Search keyword | Document collection | Document selection                  |
|--------------|----------------|---------------------|-------------------------------------|
| Wikipedia    | "S"            | Collect 1 document  | Include "S" in title or 1st paragraph |
| Naver En    | "S"            | Collect 3 document  | Distance between sentence including "S" and "P" <= 1 |
| Naver News   | "S" or "P"     | Collect 3 document  | Distance between sentence including "S" and "P" <= 1 |

For Wikipedia and Naver Encyclopedia, because the document focuses on a specific subject, the search keyword is generated using the “subject” of the user query. In contrast, Naver News is a type of document in which knowledge of various SPs is mixed. Therefore, for preventing noisy document collection, we used the search keyword generated using both “subject” and “predicate.” Furthermore, the rules for selecting the proper document from Naver News were set more strictly than those for Wikipedia and Naver Encyclopedia.

### C. SENTENCE SELECTION MODULE

The sentence selection module determines whether the input sentence is suitable for extracting the information from the user query. This module includes a proper sentence classification step that uses keyword matching and a reliability evaluation step that uses a classification model. For documents collected from Wikipedia and Naver Encyclopedia, the proper sentence is judged based on whether the sentence contains a predicate, and the reliability of the sentence is evaluated using SVM and Sentence-CNN. For documents collected from Naver News, the proper sentence is judged according to whether the sentence has both a subject and a predicate, and the reliability of the sentence is 1. When the reliability of the sentence is greater than the threshold, we select this sentence as a relevant sentence for IE.

1) SVM

The SVM model [18] calculates the confidence score of sentences classified as proper sentences. We train the SVM model using the data generated by dividing the training data by attributes during the seed data generation section. Each data is tokenized to generate a tf–idf vector and used as the input of the SVM model.

During training, the SVM model uses a binary label that contains the infobox value of the statement. Because we train the SVM model using attributes, the model is created with
the same number of attributes as in the data. However, if the quantity of data points with a specific attribute is less than 10, the model of this attribute is not created.

During testing, the predicted label score of the SVM model was used as the reliability of the sentence.

2) SENTENCE CNN
Sentence-CNN [19], which uses a simple CNN with one dimension (a convolution filter) [20], is a useful model for text classification. We used the Sentence-CNN model to calculate the confidence score of sentences classified as proper sentences. In contrast to the SVM model, which is divided by attributes, we trained a Sentence-CNN model using the complete dataset. Accordingly, the model can calculate the score regardless of the specific attribute.

During training, the Sentence-CNN model uses a binary label, based on whether it contains the value of the infobox of the sentence.

During testing, the predicted label score of the Sentence-CNN model is used as the reliability of the sentence.

D. INFORMATION EXTRACTION MODULE
The IE module extracts answers from selected sentences using the sentence selection module. The model used for IE is as follows:

**IE Model-Predicate-Based(Predicate) (IEM-PB(P))**: IEM-PB(P) is a predicate-specific IE model that learns one extractor per predicate. Each extractor is a bidirectional long short-term memory (BiLSTM) and conditional random field (BiLSTM-CRF)-based model that takes both sentence and predicate as inputs.

**IE Model-Predicate-Based(All) (IEM-PB(A))**: IEM-PB(A) is a general-purpose IE model that learns one extractor with all data. It is a BiLSTM-CRF-based model that takes both sentence and predicate as inputs.

**IE Model-SP-based(All) (IEM-SPB(A))**: IEM-SPB(A) is a general-purpose IE model that learns one extractor with all data. It is a BiDAF-based model that takes the sentence, subject, and predicate as inputs.

During the test, IEM-PB(P) operates only when the predicate of the user query matches, whereas other extractors always work on all user queries. The features used as inputs of the three extractors are presented in Table 3. Based on the input, a maximum of 9 outputs can be created from three sources (Wikipedia, Naver Encyclopedia, Naver News) and three models (IEM-PB(P), IEM-PB(A), and IEM-SPB(A)) per query. Then, the result is passed through the post-processing and knowledge ensemble modules.

1) BiLSTM-CRF(IEM-PB(P), IEM-PB(A))
IEM-PB(P) and IEM-PB(A) were designed based on the BiLSTM-CRF model. Figure 4 illustrates the BiLSTM-CRF structure used in this study. BiLSTM-CRF [21] was created by combining BiLSTM and CRF.

LSTM [22] is a modified structure of RNNs [23] that overcomes RNNs gradient vanishing and explosion problems.

LSTM is commonly used for sequential data. Recently, it has been used for many natural language processing (NLP) tasks. The mathematical representation of the LSTM model is as follows:

\[
i_t = \sigma(W_i[h_{t-1}, x_t]) + b_i \tag{1}
\]
\[
f_t = \sigma(W_f[h_{t-1}, x_t]) + b_f \tag{2}
\]
\[
o_t = \sigma(W_o[h_{t-1}, x_t]) + b_o \tag{3}
\]
\[
\tilde{c}_t = \sigma(W_c[h_{t-1}, x_t]) + b_c \tag{4}
\]
\[
c_t = i_t \ast \tilde{c}_t + f_t \ast c_{t-1} \tag{5}
\]
\[
h_t = o_t \ast \tanh(c_t) \tag{6}
\]

where \(\sigma\) is the sigmoid activation function. \(\tanh\) is the hyperbolic tangent function. \(x_t, i_t, f_t, \) and \(o_t\) are the trainable weights and biases present at each gate. \(\tilde{c}_t\) is the input of the current state, and \(c_t\) is the update state at time \(t\). \(h_t\) is the output at time \(t\). Finally, we obtain an output vector \((h_0, h_1, \ldots, h_t)\). However, LSTM only considers the forward information.

In sequence tagging, it is necessary to consider forward and backward information simultaneously.

| Feature Type        | Feature                | Description                      |
|---------------------|------------------------|----------------------------------|
| Word Vector Feature | Word embedding         | Embeddings of the word           |
|                     | Word Morpheme          | Morpheme of the word             |
|                     | Last Word              | Whether it is last word of the sentence |
| Syntactic Feature   | First Word             | Whether it is first word of the sentence |
|                     | Digit Word             | Whether it is digit word of the sentence |
| Query’s Predicate Feature | Unit Word of Query’s Predicate | Whether it is unit word of the query’s predicate |
|                     | Query’s Predicate      | Whether it is word of the query’s predicate |

![Figure 4. Architecture of BiLSTM-CRF model (IEM-PB(P), IEM-PB(A)).](image-url)
In this study, the BiLSTM model [24] was used. The BiLSTM architecture is a method of concatenating the context representation of forward-LSTM and backward-LSTM in the reverse direction. When the forward context representation vector is $\mathbf{h}_t$ and the backward context representation vector is $\overrightarrow{\mathbf{h}}_t$, the final context representation vector is $[\mathbf{h}_t; \overrightarrow{\mathbf{h}}_t]$. The final tagging can be extracted with this created context, but the dependencies between the tags are essential to the tagging problem.

We added CRF as the last layer. The CRF model [25] is designed to reflect the dependencies of adjacent labels. In this study, the CRF model was used after BiLSTM. BiLSTM reflects bidirectional context information, and CRF finds the optimal tag path from all possible tag paths that consider label dependencies. For a given sentence $X$, the quantitative definition of the probability that the prediction result is as follows:

$$s(X, \mathbf{y}) = \sum_{i=1}^{n} (T_{y_{i-1}, y_{i}} + P_{y_{i}, y_{i+1}})$$

$$p(y|X) = \frac{e^{s(X, \mathbf{y})}}{\sum_{\mathbf{y}} e^{s(X, \mathbf{y})}}$$

where $T$ represents the scores of any two adjacent labels, $T_{y_{i-1}, y_{i}}$ is the score from the successful transfer of the label $y_{i-1}$ to the label $y_{i}$, and $P_{y_{i}, y_{i+1}}$ is the confidence score of the $y_{i}$ in the label of the character $c_i$.

In the training phase, the objective function (8) maximizes the log-probability of the correct tag sequence. Then, we compute the probability and output $y$ of the label sequence using the Viterbi algorithm [26]. The output $y$ of the label sequence is as follows:

$$y = \arg\max_{\mathbf{y}} s(X, \mathbf{y})$$

2) BiDAF(IEM-SPB(A))

The IEM-SPB(A) model was designed by modifying BiDAF and can reflect both the input context and user query. Both input context and user query have been encoded using the general BiLSTM structure, and the two are fused using the Attention Flow Layer of BiDAF. The Attention Flow Layer obtains the attention weight in the Context direction from the query and generates the Query2Context vector based on a weighted sum with the context vector sequence. In the opposite direction, the Attention weight is calculated from the context to query the direction for each step of the statement, and a weighted sum with the query vector sequence is calculated to obtain the Context2Query vector for each step. Before calculating the attention weight, we first calculate the similarity matrix as follows:

$$S_{ij} = \alpha(H_{it}, U_{jt})$$

where $\alpha$ is a trainable scalar function that encodes the similarity between its two input vectors, $H_{it}$ is the $t$-th column context vector of $\mathbf{H}$, $U_{jt}$ is the $j$-th column vector of $\mathbf{U}$, $\alpha(h, u) = w^T_S[h; u; h \circ u]$, where $w_S$ is a trainable weight vector, $\circ$ is elementwise multiplication, and $[;]$ is vector concatenation across the row.

The Context2Query attention is:

$$a_t = \text{softmax}(S_t)$$

$$\tilde{U}_{it} = \sum_j a_{ij} U_{jt}$$

The Query2Context attention is:

$$b = \text{softmax}(\max_{col}(S))$$

$$\mathbf{h} = \sum_t b_t H_{it}$$

where $\mathbf{h}$ is tiled $T$ times across the column, thus producing $\mathbf{H}$. Moreover, the contextual embeddings and the attention vectors are combined to yield $\mathbf{G}$, defined by:

$$G_{it} = \beta(H_{it}, \tilde{U}_{it}, \mathbf{h})$$

where $G_{it}$ is the $t$-th context vector, and $\beta$ is a simple concatenation: $\beta(h, \tilde{u}, \mathbf{h}) = [h; \tilde{u}; h \circ \mathbf{h}]$. The attention vector $G$ is used as the input to the BiLSTM layer. Then, we used a fully-connected layer and the softmax function to predict the final label. Let $M = (m_1, m_2, \ldots, m_t)$ be the output vector of the BiLSTM layer. Then, the final label $m_t$ at time $t$ is calculated as follows:

$$y_t = \text{softmax}(W m_t + b)$$

where $W$ and $b$ are trainable weights and biases for the softmax layer. Figure 5 illustrates the structure of the model used in this study.

3) POST-PROCESSING

Post-processing used to supplement the information derived based on the predefined unit dictionary before combining the results extracted from the three models. For example, if you obtain a result of “41,000” for an input such as (“the Great
Wall," "length")}, the post-processing module adds "km" units to the output and changes the result to "41,000km."

This approach only succeeds when the result is included in the unit information and does not add any additional information. Information about the query output from the IE model is returned through the post-processing module when the unit information is missing.

In this study, the response unit information for the predicate was defined in advance for 10 query predicate categories (e.g., length, weight, speed, and size).

4) KNOWLEDGE ENSEMBLE
The knowledge ensemble is the final step in generating the final answer based on the results of the IE module and post-processing. In this step, two knowledge ensemble methods were performed.

The first was the simple soft computing method—Simple Ensemble Knowledge Extraction Model (SEM)—which summates scores by answer and extracts the answer with the highest score as the final knowledge. Because no additional learning is required, very few resources are needed, and the final knowledge can be effectively extracted.

The second method—a Neural Ensemble Knowledge Extraction Model (NEM)—uses neural networks as a predicate-based neural weight summation method. NEM is designed with the assumption that each knowledge extractor performs more accurately for a particular predicate. In the neural network, the knowledge score extracted from each model and predicate information are input to generate a new score that reflects the weight of each model. Then, the score is summed by knowledge to extract the knowledge with the highest score as the final knowledge. A total of 10,000 pairs of queries and labels were sampled in the seed data to train the model.

The performance of the proposed system was evaluated for 400 test queries—for the positive condition, negative condition, and accuracy. If the IE score is lower than the threshold, the model does not extract the answer. The default threshold for IE models is 0.5. We evaluated our proposed models—

IV. EXPERIMENTS
A. DATASET
For measuring the performance of our IE system, SKTelecom was provided with a portion of the query requesting the property of an entity among the actual user queries of the AI speaker. Each query is divided into SP entity-property, approximately 200,000 queries. However, because a person must directly annotate, we selected 400 test queries by random sampling. And we built a test dataset by collecting approximately 2,800 unstructured documents from multiple Web sources for 400 test queries. An example of the Korean user query log of the SKTelecom AI speaker is presented in Table 4.

B. EVALUATION METRICS
In this study, the precision, recall, and F1 score for each positive and negative condition were calculated to produce a quantitative value. Moreover, the performance was evaluated by deriving accuracy to confirm the overall performance of the system.

C. EVALUATION
The performance of the proposed system was evaluated for 400 test queries—for the positive condition, negative condition, and accuracy. If the IE score is lower than the threshold, the model does not extract the answer. The default threshold for IE models is 0.5. We evaluated our proposed models—
TABLE 6. IE performance evaluation formula.

| Metric       | Formula                     |
|--------------|-----------------------------|
| Precision$_{POS}$ | $TPC/TPC+FP+FN$              |
| Recall$_{POS}$  | $TPC/TPC+FN$                |
| F1-score$_{POS}$| $2 \times \text{Precision}_{POS} \times \text{Recall}_{POS} / \text{Precision}_{POS} + \text{Recall}_{POS}$ |

TABLE 7. Model performance in Wikipedia.

| Model          | Positive |          |          |        |
|----------------|----------|----------|----------|--------|
|                | P        | R        | F        | Acc    |
| KYLIN          | 83.1     | 22.1     | 33.0     | 64.0   |
| BiDAF          | 4.6      | 8.2      | 5.9      | 18.2   |
| SEM            | 65.3     | 63.5     | 64.4     | 79.5   |
| NEM            | 70.2     | 69.0     | 69.6     | 82.2   |
| NEM(0.9)       | 91.5     | 48.0     | 63.0     | 75.7   |

TABLE 8. Model performance in Naver Encyclopedia.

| Model          | Positive |          |          |        |
|----------------|----------|----------|----------|--------|
|                | P        | R        | F        | Acc    |
| KYLIN          | 84.6     | 20.3     | 32.3     | 56.7   |
| BiDAF          | 3.4      | 6.0      | 4.3      | 7.2    |
| SEM            | 61.0     | 60.1     | 60.6     | 74.0   |
| NEM            | 64.4     | 63.8     | 64.1     | 76.5   |
| NEM(0.9)       | 87.0     | 46.7     | 60.8     | 70.2   |

SEM, NEM, and NEM(0.9)—with a threshold of 0.9. Furthermore, BiDAF and KYLIN were compared with the proposed model as baselines.

Tables 7, 8, and 9 present the performance of each source (Wikipedia, Naver Encyclopedia, and Naver News). Table 10 illustrates the performance of each model for all sources.

Table 7 presents the results from Wikipedia. In positive condition, KYLIN works only on a specific predicate, so it has high precision but low recall. For BiDAF, there is no procedure for judging whether a correct answer is included in a sentence. Consequently, attempts to extract information from documents that do not have correct answers frequently result in low performance. The proposed models SEM, NEM, and NEM(0.9) improve the F1 score by 29.4, 34.6, and 28.0, compared with KYLIN. Furthermore, for the negative condition and the overall performance accuracy, the proposed models also show high performance.

For the Naver encyclopedia in Table 8, the overall performance is similar to that of Wikipedia because Wikipedia and Naver encyclopedias have the same format as an encyclopedia. Compared with KYLIN, the proposed models SEM, NEM, and NEM(0.9) improve the F1 score by 27.8, 31.3, and 28.0 for the positive condition, and by 17.3, 19.8, and 13.5 for the negative condition.

Table 9 presents the performance for the Naver News source. KYLIN exhibits very low recall for the positive condition because KYLIN extracts answers only from data similar to the training sources and does not extract answers from heterogeneous data such as news sources. Compared with the baselines, our proposed models demonstrate higher performance for all metrics. The results confirm that the proposed models in this study work well with heterogeneous documents.

Table 10 presents the performance of the total source. From the results of the experiment, the proposed models achieved significant improvements for both conditions. Compared with KYLIN, SEM improved the F1 score for the positive and negative conditions by 38.3 and 21.2, NEM improved the performance by 44.3 and 21.2.

For AI speakers in the field, it is dangerous to deliver incorrect results to the user. Therefore, it is essential to maintain high precision for the positive condition and high recall for the negative condition. For KYLIN, the performance was 85.9 (49/57) and 100 (126/126). However, KYLIN answered only 57 out of 400 queries. Therefore, the accuracy is very low, at 43.7 (175/400). In contrast, for NEM(0.9), the precision for the positive condition, recall for the negative condition, and accuracy reached 90 (99/110), 96 (121/126), and 55 (220/400). These results represent high performance for practical use in AI speakers.

Furthermore, in SQuAD 2.0 [27], a public dataset containing unanswerable questions, the F1 scores of the latest models QANet and DocQA [27] are 53.2 and 67.6, and the accuracies are 56.9 and 65.1. Although these conditions differ from those of the dataset used in the proposed model, the NEM model demonstrates higher performance despite the results in the open domain.

These results confirm that 1) the integration of the document selection module and sentence selection module increased the reliability of the answer, and 2) even if the correct answer does not exist in the document, the proposed model performs accurately.

D. SAMPLE QUERY TEST

For evaluating the performance of the proposed system in this study, 2,000 sample queries were additionally extracted from queries provided by SKTelecom to evaluate the accuracy of each threshold. A total of 1,111 answers were extracted from 2,000 sample user queries, with 713 correct answer queries. Table 12 presents the accuracy of the proposed system by threshold.
Similar to the previous 400 query tests, when the threshold was changed to low (default) and high (0.9), the accuracy was 64.17 and 94.30, respectively. Furthermore, the average time required to extract answers for each query in the proposed system is 9 seconds, and the general time required for excluding outliers is within 5 seconds. Based on these results, we can estimate the generalization performance of this system; the possibility of application to commercial systems was confirmed.

Furthermore, for approximately 200,000 Korean user queries provided by SKTelecom, we have completed collecting answers. Although we did not verify whether all answers were correct, based on the results of the sample query test, we believe that the answers are meaningful data and can be used in other systems.

V. CONCLUSION AND FUTURE WORKS
In this study, we proposed a novel IE system that can respond practically to open-domain queries, including unanswerable questions. With the proposed system, we alleviate the low-performance problem in terms of specificity, which is likely to occur when previous approaches answer open-domain unanswerable questions based on unstructured Web environments. We validated our approach by constructing an evaluation dataset by annotating the Korean user query dataset of the SKTelecom AI speaker and confirmed the effectiveness of the knowledge extraction of the proposed system. However, opportunities remain to improve the performance of the system and address limitations.

In future work, we plan to improve our system’s performance in the two directions. First, we expect additional benefits from using ELMO [28] and BERT, a recent pre-trained language model. Second, a methodology to select or expand a new knowledge source is likely to produce a correct answer when it is impossible to extract the correct answer from the given web sources.

REFERENCES
[1] T. Heath and C. Bizer, “Linked data: Evolving the Web into a global data space,” Synth. Lectures Semantic Web, Theory Technol., vol. 1, no. 1, pp. 1–136, Feb. 2011.
[2] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ questions for machine comprehension of text,” 2016, arXiv:1606.05280. [Online]. Available: http://arxiv.org/abs/1606.05280
[3] F. Wu and D. S. Weld, “Autonomously semantifying wikipedia,” in Proc. 16th ACM Conf. Conf. Knowl. Manage. CIKM, 2007, pp. 41–50.
[4] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, “Bidirectional attention flow for machine comprehension,” 2016, arXiv:1611.01603. [Online]. Available: http://arxiv.org/abs/1611.01603
[5] R. Gaizauskas and Y. Wilks, “Information extraction: Beyond document retrieval,” J. Document., vol. 54, no. 1, pp. 70–105, Mar. 1998.
[6] L. Eikvil, “Information extraction from world wide Web—A survey,” Norwegian Computing Center, Oslo, Norway, Tech. Rep. 945, 1999.
[7] T. Khot, A. Sabharwal, and P. Clark, “Answering complex questions using open information extraction,” 2017, arXiv:1704.05572. [Online]. Available: http://arxiv.org/abs/1704.05572
[8] M. Banko, M. J. Cafarella, S. Soderland, and O. Etzioni, “Open information extraction from the Web,” in Proc. Icasi, vol. 7, 2007, pp. 2670–2676.
[9] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, “Web-scale information extraction in knowItAll: (Preliminary results),” in Proc. 13th Conf. World Wide Web, 2004, pp. 100–110.
[10] S. Liu, X. Zhang, S. Zhang, H. Wang, and W. Zhang, “Neural machine reading comprehension: Methods and trends,” Appl. Sci., vol. 9, no. 18, p. 3698, Sep. 2019.
[11] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, “Teaching machines to read and comprehend,” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 1693–1701.
[12] S. Wang and J. Jiang, “Machine comprehension using match-lstm and answer pointer,” 2016, arXiv:1608.07905. [Online]. Available: http://arxiv.org/abs/1608.07905
[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805
[14] A. Wei Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, “QANet: Combining local convolution with global self-attention for reading comprehension,” 2018, arXiv:1804.09541. [Online]. Available: http://arxiv.org/abs/1804.09541
[15] A. Vaswani, N. Shazeer, N. Parmar, J.Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.
[16] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman, “NewsQA: A machine comprehension dataset,” 2016, arXiv:1611.09830. [Online]. Available: http://arxiv.org/abs/1611.09830
[17] M. Richardson, C. J. Burges, and E. Renshaw, “MCTest: A challenge dataset for the open-domain machine comprehension of text,” in Proc. Conf. Empirical Methods Natural Lang. Process, Seattle, WA, USA: Association for Computational Linguistics, Oct. 2013, pp. 193–203.
[18] T. Joachims, “Text categorization with support vector machines,” in Proc. Eur. Conf. Mach. Learn., vol. 7, 2007, pp. 137–142.
[19] Y. Kim, “Convolutional neural networks for sentence classification,” 2014, arXiv:1408.5882. [Online]. Available: http://arxiv.org/abs/1408.5882
[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.
[21] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” 2015, arXiv:1508.01991. [Online]. Available: http://arxiv.org/abs/1508.01991

[22] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[23] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, “Recurrent neural network based language model,” in Proc. 11th Annu. Conf. Int. Speech Commun. Assoc., 2010, pp. 1045–1048.

[24] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Trans. Signal Process., vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[25] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” 2018, arXiv:1802.05365. [Online]. Available: http://arxiv.org/abs/1802.05365

YEONGTAEK OH received the M.S. degree in industrial engineering from Yonsei University, in 2019. He is currently a Data Scientist with SK Hynix. His main research interests include semiconductor manufacturing process, natural language processing, and machine learning.

SANGHEON LEE received the Ph.D. degree in industrial engineering from Yonsei University, in 2019. He is currently a Data Scientist with SK Hynix. His main research interests include dialog system, natural language processing, and machine learning.

YEONGTAEK OH received the M.S. degree in industrial engineering from Yonsei University, in 2019. He is currently a Data Scientist with SK Hynix. His main research interests include semiconductor manufacturing process, natural language processing, and machine learning.

SANGHEON LEE received the Ph.D. degree in industrial engineering from Yonsei University, in 2019. He is currently a Data Scientist with SK Hynix. His main research interests include dialog system, natural language processing, and machine learning.

HYUNSEUNG CHOI received the M.S. degree in industrial engineering from Yonsei University, in 2019. He is currently a Data Scientist with SK Hynix. His main research interests include artificial intelligence technology, natural language processing, and computer vision.

WOOJU KIM received a Ph.D. degree in Operation Research from KAIST, Korea, in 1994. He is currently a Professor at the School of Industrial Engineering, Yonsei University. His main research interests include natural language processing, reliable knowledge discovery, big data intelligence, machine learning, and artificial intelligence.

* * *