IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 2024, date of current version Oct. 15, 2024.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

End-to-End Trainable Retrieval-Augmented Generation for Relation Extraction

KOHEI MAKINO, MAKOTO MIWA, and YUTAKA SASAKI
Toyota Technological Institute, 2-12-1 Hisakata, Tempaku-ku, Nagoya, 468-8511 Japan
Corresponding author: Yutaka Sasaki (e-mail: yutaka.sasaki@toyota-ti.ac.jp).

ABSTRACT This paper addresses a crucial challenge in retrieval-augmented generation-based relation extractors; the end-to-end training is not applicable to conventional retrieval-augmented generation due to the non-differentiable nature of instance retrieval. This problem prevents the instance retrievers from being optimized for the relation extraction task, and conventionally it must be trained with an objective different from that for relation extraction. To address this issue, we propose a novel End-to-end Trainable Retrieval-Augmented Generation (ETRAG), which allows end-to-end optimization of the entire model, including the retriever, for the relation extraction objective by utilizing a differentiable selection of the k nearest instances. We evaluate the relation extraction performance of ETRAG on the TACRED dataset, which is a standard benchmark for relation extraction. ETRAG demonstrates consistent improvements against the baseline model as retrieved instances are added. Furthermore, the analysis of instances retrieved by the end-to-end trained retriever confirms that the retrieved instances contain common relation labels or entities with the query and are specialized for the target task. Our findings provide a promising foundation for future research on retrieval-augmented generation and the broader applications of text generation in Natural Language Processing.

INDEX TERMS Differentiable retrieval, End-to-end training, Relation extraction, Retrieval-augmented generation

I. INTRODUCTION

Relation extraction is a fundamental task in Natural Language Processing (NLP), which involves identifying and classifying semantic relationships between entity mentions, such as people, organization, and location names, in text [1]. Relation extraction plays a crucial role in understanding and interpreting the underlying meaning of sentences by analyzing how entities are interrelated [2], [3]. [1]

Relation extraction is used in practical applications in several domains. For instance, in knowledge graph construction, relation extraction aids in transforming unstructured text into structured data, which can then be used to populate and enrich knowledge graphs [4]. In domain-specific scenarios, such as biomedical text mining [5], it can extract relationships between genes, diseases, and drugs, which are helpful for advanced research and discovery such as search systems [6] and prediction of novel things [7]. Thus, relation extraction is used for a wide variety of purposes.

Developing relation extractors is an ongoing endeavor in NLP. Relation extractors are designed to accurately identify and classify relations in text, which requires understanding the nuanced and sometimes complex language structures. Advances in machine learning and NLP methods have shaped the evolution of the extractors. Recent extractors are based on deep learning models to obtain high-performance [8]–[10], while traditional extractors are rule-based models [11] and feature-based models [12].

The Pretrained Language Models (PLMs) [13]–[15] have become a de facto standard in recent NLP, fundamentally changing the field landscape. PLMs are neural network models pretrained on a common NLP task, such as language modeling [16] and masked language modeling [14], and are used by fine-tuning it to fit a target task. Many studies on relation extraction with PLMs have been conducted because of the high performance [17], [18].

With the advent of PLMs, well-trained text generation models, including Large Language Models (LLMs) [19]–[21] trained on a larger corpus with larger-scale parameters, are attracting attention [16], [22], [23]. Text generation model-based relation extractors [24]–[26] are used because

[1] This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

the pretraining and the fine-tuning are similar and it helps the extractor training. These models have been adapted to treat relation extraction as a question-answering task [24], a text summarization task [27], or a language modeling [25], leveraging their knowledge and understanding of natural language.

Some existing works utilize relation instances, which consist of text and relation between entities, to improve the performance. Instance-based methods perform better in low-resource settings with insufficient training data because they can use known examples as anchors. A typical instance-based extractor using PLM predicts the label of the nearest known instance on the encoded representation [28]. However, the inference with simple voting with nearest neighbor methods is weak in terms of the overall performance compared to other models.

Combining text generation models with instance utilization has been particularly effective with In-Context Learning (ICL) [19]. In relation extraction via ICL, instructions and few-shot instances verbalized as text are used as prompts to supervise text generation models from context, and the model generates the text convertible to relation labels conditioned by the prompt [25]. ICL, which predicts based on context, can produce output using instances as hints rather than methods such as the nearest neighbor method. Since standard ICL uses pre-defined instances in the prompt, ICL methods cannot access instances adaptable to input.

To solve these problems, Retrieval-Augmented Generation (RAG) [29] introduces the instances relevant to the input into the text prompt instead of pre-defined instances. In RAG-based relation extractors [26], retrievers prepared separately from the base model select instances. The retrievers use the target text as a query and select instances from the relation instances stored as a database. The selected instances are then introduced into the text prompt. Generally, the retrieval is done by embedding the inputs and instances into a common dense feature space by a separately trained encoder model and selecting the nearest neighbors. Thus, RAG introduces adaptive instances to the prompt of the text generation model.

The retriever used in RAG requires training to retrieve the appropriate instance when used in the target task. However, the instance retriever needs to be trained separately from the target relation extraction task because the text generation model with the retriever cannot be trained end-to-end. In other words, designing a retriever for each target problem is necessary, which increases the development cost [26]. Therefore, end-to-end training allows direct optimization of the retriever to the target task, reducing development costs by eliminating the need to devise case-specific training methods.

Our goal is to create an environment where the entire model with RAG can be optimized for the relation extraction task. Therefore, this study aims to make the RAG model end-to-end trainable. Specifically, we eliminate indifferentiable operations in the retriever that prevent the training of deep learning models by replacing them with differentiable operations. By integrating the operations, we propose ETRAG (End-to-end Trainable Retrieval-Augmented Retrieval) as a RAG that can train the entire model end-to-end. Our proposed method is evaluated by a relation extraction task to confirm effectiveness and characteristics.

Contributions in this paper are threefold.

- We propose end-to-end trainable RAG, ETRAG, which replaces the k-nearest neighbor method with differentiable operations and uses obtained instances as soft prompts. ETRAG enables an entire model, including the retriever, to fine-tune for the relation extraction task.
- We confirm that ETRAG consistently improves extraction performance for the TACRED, a benchmark for relation extraction, and is particularly effective in situations where training data is limited.
- Our analysis reveals that ETRAG can select instances strongly related to the target task. For the relation extraction, instances with the same relationship labels as the extraction target and instances containing the same surface entities account for more than 70% of the instances.

The remainder of this paper is organized as follows: Section II presents related work, further elaborating on the evolution and current state of PLM (Section II-A), instance-based methods (Section II-B), and relation extraction (Section II-C). Section II-C contains notations and explanations used in subsequent sections. Section III explains our methodology by describing the proposed method ETRAG with differentiable k-nearest neighbor (kNN) (Section III-A) and neural prompting (Section III-B), and the training methods of ETRAG (Section III-C). The evaluations are performed in Section IV for the extraction performance and Section V for the retriever analysis. Finally, Section VI concludes this study and describes future directions.

II. RELATED WORK

A. PRETRAINED LANGUAGE MODELS

PLMs are large-scale neural network models trained on a large corpus with NLP tasks and are usually fine-tuned when applied to various tasks. Pretraining tasks are extensive such as language modeling (LM) [16], bidirectional language modeling (biLM) [13], sequence-to-sequence language modeling (Seq2Seq LM) [22], and masked language modeling (MLM) [14]. PLMs employ appropriate neural network structures such as ELMo [13] for the bidirectional language modeling task with LSTMs [30], and Transformer [31] is mainly used recent systems including BERT [14] for masked language modeling, T5 [22] for sequence-to-sequence language modeling, and GPT-2 [16] for language modeling. PLMs are extended to a larger scale and use a larger corpus such as Flan-T5 [20] in Seq2Seq LM and GPT-3 in LM [19].

PLMs are stochastic models to estimate the probability of word sequences based on the pretraining tasks; collectively, these are called text generation models G. Since LM is a task to predict the subsequent text from a given input, the model trained on LM can estimate the probability of input text sequence, formulated as $G(x)$ with input x. On the other hand, the model trained on Seq2Seq LM can assign a probability to input-output pairs used in pretraining, formulated as $G(y|x)$ with input x and output y.

PLMs are computationally expensive to fine-tune the entire parameters when adapted to the target task. Thus, they are tuned by controlling the generated text with text prompts or by parameter-efficient tuning. ICL performs the target task learned from the context of tuned text prompts [19]. However, the performance on the target task is often lower than that of a model tuned specifically to the target task, and the prompt tuning process depends on the expert. Instead of the text prompt, prompt tuning [32] or prefix-tuning [33] trains soft prompts, which are trainable vectors inserted to embedded text sequence. Neural prompting [34] has extended soft prompts to create them by encoding external information rather than pre-prepared embeddings. Since such prompt-based tuning can tune the context but not the model parameters, lightweight model tuning can also be used [35], [36]. Low-Rank Adaptation (LoRA) [36] inserts low-rank parameters into the base model to learn the difference between the target task. We utilize the LoRA in our experiments to update PLM and the neural prompting in our method to introduce instances.

B. RETRIEVAL-BASED METHODS

Retrieval-based methods, as typified by a nearest neighbor method [37], have been used for various NLP tasks, such as Part-Of-Speech (POS) tagging [38], named entity recognition [39], dependency parsing [40], and relation extraction [41]. The methods are used to mitigate a training data scarcity situation. This paper replaces the indifferentiable operations in the kNN algorithm with differentiable ones in order to relax to a soft operation.

Recently, PLMs are often used following the pretraining task to use instances by ICL, which predicts the following context from the given prompt. Since ICL is characterized by the prompt design, the instance selection plays a vital role. Since fixed prompts are usually used, the instances are also typically fixed. However, the demand to select the most appropriate instances for input has led to use RAG [29], which trains a retriever in advance and uses the instances obtained by the retriever. Since the instance selection operation is not differentiable because of sampling, the general retriever is trained separately to the target task [26]. This study tackles this separate training of the target task model and the retriever so that it can be trained end-to-end with the target task.

C. RELATION EXTRACTION

The fundamental relation extraction task is sentence-level relation extraction [42], [43], where only the entity pairs in a single sentence are targets for extraction. Since sentence-level relation extraction ignores relations across sentences, it is extended to document-level relation extraction [44], [45], which also extracts these relations. Since this paper focuses on a retriever that retrieves instances tied to a single relation for simplicity, we develop an extractor primarily on sentence-level relation extraction.

Historically, relation extractors have evolved from traditional approaches to modern deep learning techniques. Initially, rule-based systems relied on hand-crafted rules to identify relations [11]. Kernel-based models and feature-based approaches later emerged, offering more flexibility and better handling linguistic variations [12], [46]. Deep learning revolutionized the field, introducing models that could learn complex patterns and relationships directly from data [47].

Given input sentence x with head entity h and tail entity t of a target entity pair and relation $r \in R$ between them, a standard relation extractor in Equation (1) formulates a classification task using a stochastic model P which is constructed with deep learning.

$$\hat{r} = \operatorname*{argmax}_{r \in R} P(r|x, h, t) \tag{1}$$

1) Relation Extraction with Pretrained Language Models

Most recent relation extractors employ PLMs for modeling P as feature extractors to prepare the feature vector for direct classification [48] or prepare the features for the following relation extraction-specific model [49]. On the other hand, PLMs are also used as the text generation model with prompt engineering [26] or fine-tuning [25]. For example, SuRE (Summarization as Relation Extraction) [27] extracts the relation with summarization via PLM and mapping the summarized text output to relations. SuRE measures the probability of a pair of input text prompts and verbalizes relations in a summary form to predict the relation with the highest probability.

Typical text generation-based extractors prepare input and output templates with placeholders to fill with information to ask about relations. Let an input template be $\mathrm{Template}_{\mathrm{in}}(x, h, t)$, an output template be $\mathrm{Template}_{\mathrm{out}}(x, h, t, r)$. Equation (2) and Equation (3) show examples of them.

$$\begin{aligned}\mathrm{Template}_{\mathrm{in}}(x, h, t) = \text{“The head entity is } h\text{ .}\\ \text{The tail entity is } t\text{ . } x\text{”}\end{aligned} \tag{2}$$

$$\begin{aligned}\mathrm{Template}_{\mathrm{out}}(x, h, t, r = \text{no\_relation}) =\\ \text{“}h\text{ has no known relations to } t\text{ .”}\end{aligned} \tag{3}$$

Let a text generation model that computes the probability of the output template conditioned by the input sequence be $G(\mathrm{Template}_{\mathrm{out}} \mid \mathrm{Template}_{\mathrm{in}})$, Equation (4) represents the text generation-based relation extractor.

$$\hat{r} = \operatorname*{argmax}_{r \in R} G(\mathrm{Template}_{\mathrm{out}}(x, h, t, r) \mid \mathrm{Template}_{\mathrm{in}}(x, h, t)) \tag{4}$$

SuRE is a typical method for relation extraction in this form, which improves the efficiency of text-generation-based relation extraction according to Equation (4). Calculating probabilities for all relation labels in a straightforward manner is expensive due to the large number of calculations required by PLM. Therefore, SuRE prepares a trie tree of templates for all relation labels and searches for the template with the highest probability by beam search on the trie, thereby realizing the prediction of relations by text generation with a small number of calculations. Note that the model training is the same as that of normal text generation models.

2) Relation Extractor Using Instances

Despite recent advancements in deep learning models, relation extraction remains challenging, primarily due to the scarcity of annotated data. Deep learning models, in particular, require large amounts of labeled data for training [8]. However, manually annotating data is expensive and time-consuming, making it a significant bottleneck in developing effective relation extraction systems [50]. This environment makes it difficult to classify with high performance for complex relations types. In this context, the relation extractor via the nearest neighbor approach [28], where relation extractors leverage similar instances during inference, has shown promise in efficiently utilizing limited data. The goal of this study is to reveal that using instances, or specific instances of relations within texts, is crucial in relation extraction. By leveraging these instances, models can better generalize from limited instances and improve their ability to extract and classify relations in varied contexts accurately.

The relation extractors with the text generation model also use instances by introducing selected instances as text into the prompt [25], [26]. Such methods extend the templates to accept selected instances so that the text generation model can handle instances by preparing placeholders for them. When using a retriever [26], instances are prepared according to the input, and when not using a retriever [25], instances are prepared in advance manually.

The instances enhanced text generation model involves two processes: the retrieved instances into prompts compatible with the text generation model. In the traditional RAG framework [29], the retriever identifies neighboring instances and directly utilizes the text of these instances as prompts. Conversely, in neural prompting [34], instances are selected based on string matching and processed through a neural network to use the instances as soft prompts, offering a more nuanced and adaptable approach to instance utilization. Let an external database for a reference, which includes the elements of a text and relation information (x, h, t, r), be D, a set of selected instances be $D' \subseteq D$, the template that accepts instances as input be $\mathrm{Template}_{\mathrm{in}}(x, h, t, D')$, and the retrieval process to obtain D' from D using input information be $D' = \mathrm{Retrieve}(x, h, t, D)$, the text generation-based relation extractor with instances are shown in Equation (5).

$$\hat{r} = \operatorname*{argmax}_{r \in R} G(\mathrm{Template}_{\mathrm{out}}(x, h, t, r) \mid \mathrm{Template}_{\mathrm{in}}(x, h, t, D')) \quad (5)$$

In the retriever that searches nearest instances in the embedding space, each instance d in the database D is converted into L embeddings E_d:

$$E_d = [E_{1,d}, E_{2,d}, \ldots, E_{L,d}]^\top \in \mathbb{R}^{L \times N}. \quad (6)$$

The retriever aims to select instances that are near the input embedding $E_{\mathrm{in}} \in \mathbb{R}^{L \times N}$, which is embedded similarly to E_d using x, h, and t. The method to embed instances and the input is usually engineered to suit the purpose; for example, modern applications [51] often use PLMs such as BERT [14]. Our method also uses PLM to embed instances. Nearest instance search requires the distance between input and each instance calculated with a function to compute distance as $\mathrm{Dist}(E_{\mathrm{in}}, E_d)$. Since the kNN retriever's objective is to emit instances within a distance of up to k-th, a set of target instances D' becomes Equation (7), where $\mathrm{argTopK}$ returns a set of top-k indices:

$$D' = \left\{ D_i \mid i \in \operatorname*{argTopK}_{j} -\mathrm{Dist}(E_{\mathrm{in}}, E_j) \right\} \quad (7)$$

After the retrieval, the instances are converted into a format that can be input into the model in the embedding process, e.g., the traditional method writes down into text prompts for PLM [29].

III. METHODOLOGY

We propose ETRAG, which enables end-to-end training of text generation models with RAG. In order to fit a model pretrained for a general task to the target task, the parameters need fine-tuning to the objective of the target task. However, the retriever part of general RAG cannot be trained end-to-end, and the instances selection cannot be optimized when training the model on the target task. The entire model must consist of differentiable operations to compute gradient when training a deep learning model end-to-end. However, two indifferentiable operations prevent end-to-end training: selecting instances by retriever and making the selected instances into a text prompt.

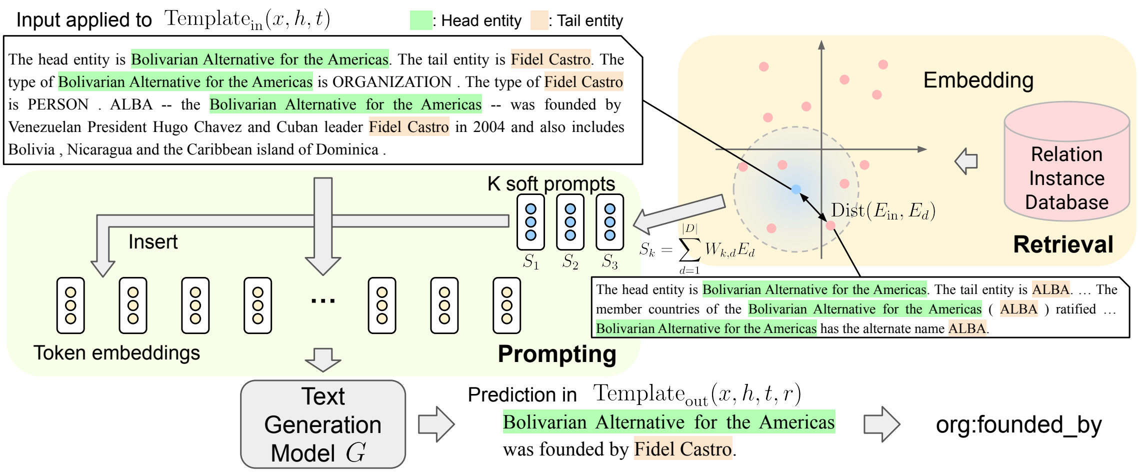
To overcome the indifferentiable processes, ETRAG replaces the retriever's process of selecting instances with a soft kNN and introduces instances as soft prompts as shown in Figure 1. For the retriever selection, we employ a soft kNN, which selects instances softly inspired by neural nearest neighbor networks [52]. When introducing selected instances, ETRAG injects the instances to input as soft prompts, which concatenates instance embeddings into embedded input tokens rather than text prompts, like neural prompting [34].

To extract relations by a text-generating model, we employ SuRE [27] as the base relation extractor. SuRE is a method of inference by a text-generating model, and since the model is a regular text-generation model, we propose a method for introducing instances for the text-generation model.

We will explain the method in the following sections: ETRAG consisting of differentiable k-nearest instance selection (Section III-A) and integration of the instances drawn by it (Section III-B). The end-to-end training and training techniques for ETRAG are shown in Section III-C.

A. DIFFERENTIABLE *K*-NEAREST INSTANCE SELECTION

The differentiable k-nearest instance selection is achieved by weighted selection over multiple instances. In contrast,


Input applied to $\mathrm{Template_{in}}(x, h, t)$
🟩 : Head entity
🟧 : Tail entity
Embedding
Relation Instance Database
Retrieval
$\mathrm{Dist}(E_{\mathrm{in}}, E_d)$
K soft prompts
S_1 S_2 S_3
$S_k = \sum_{d=1}^{|D|} W_{k,d} E_d$
Insert
Token embeddings
Prompting
Text Generation Model G
Prediction in $\mathrm{Template_{out}}(x, h, t, r)$
org:founded_by

The head entity is Bolivarian Alternative for the Americas. The tail entity is Fidel Castro. The type of Bolivarian Alternative for the Americas is ORGANIZATION . The type of Fidel Castro is PERSON . ALBA -- the Bolivarian Alternative for the Americas -- was founded by Venezuelan President Hugo Chavez and Cuban leader Fidel Castro in 2004 and also includes Bolivia , Nicaragua and the Caribbean island of Dominica .

The head entity is Bolivarian Alternative for the Americas. The tail entity is ALBA. … The member countries of the Bolivarian Alternative for the Americas (ALBA) ratified … Bolivarian Alternative for the Americas has the alternate name ALBA.

Bolivarian Alternative for the Americas was founded by Fidel Castro.

FIGURE 1. An overview of ETRAG

the standard k-nearest instance selection is indifferentiable because instances are selected with sampling operation as shown in Equation (7). Therefore, in this section, we first describe the embedding of instances created by weighted sums of the instances and then explain the creation of weights.

We formulate the differentiable instance selection by creating a weighted sum of the instance embeddings, paying attention to the important instances. Let weights be $W \in \mathbb{R}^{K \times |D|}$, where $W_{k,d}$ means the weight of d-th data for k-th selection, k-th selected instance embedding $S_k \in \mathbb{R}^{L \times N}$ is defined as $S_k = \sum_{d=1}^{|D|} W_{k,d} E_d$.

We then explain how to compute the weights in a differentiable manner. The simplest solution to sample the nearest instance is applying softmax to the distance between instances such as Gumbel softmax [53]. However, this approach is possible to select only the nearest neighbor instance, not the second, third, and subsequent instances. In order to realize subsequent selection, instances with heavy weights in previous steps are penalized, and the weights in subsequent steps are computed based on them so that the instances once heavily weighted process almost exclusively, as described in Equation (8).

$$W_{k,d} = \begin{cases} \frac{\exp(1 - \mathrm{Dist}(E_{\mathrm{in}}, E_d))}{\sum_{d'=1}^{|D|} \exp(1 - \mathrm{Dist}(E_{\mathrm{in}}, E_{d'}))} & \text{if } k = 1 \\ \frac{\exp(1 - \mathrm{Dist}(E_{\mathrm{in}}, E_d) + \sum_{l=1}^{k-1} \log(1 - W_{l,d}))}{\sum_{d'=1}^{|D|} \exp(1 - \mathrm{Dist}(E_{\mathrm{in}}, E_{d'}) + \sum_{l=1}^{k-1} \log(1 - W_{l,d'}))} & \text{otherwise} \end{cases} \tag{8}$$

The weights W_k become a nearly one-hot vector since the softmax function computes the selection weight. We assume the distance $\mathrm{Dist}(E_{\mathrm{in}}, E_d)$ is an average of the cosine distance between E_{in} and E_d as defined in Equation (9), where Dist is bounded from 0 to 1 ($0 \leq \mathrm{Dist} \leq 1$).

$$\mathrm{Dist}(E_{\mathrm{in}}, E_d) = \frac{1}{2L} \sum_{l=1}^{L} 1 - \frac{E_{\mathrm{in},l}^{\top} E_{d,l}}{|E_{\mathrm{in},l}||E_{d,l}|} \tag{9}$$

The retrieval for relation extraction requires the preparation of embeddings E. We use representations of head and tail entities and relation labels, which are considered helpful for relation extraction [54]. These representations are obtained by writing down the relation instances in text using templates, embedding them using another PLM, and extracting the representations of the text corresponding to the head entity, tail entity, and relation labels. Specifically, we use an average of embeddings in the span of entities or relations, where $E_{\cdot,1}$ and $E_{\cdot,2}$ are the embeddings of head and tail entity and $E_{\cdot,3}$ is the embedding of relation.

These processes provide K weights for each instance and K selected instances. When using this k-nearest instance selection, the neural model constructed using the weights to select instances can select instances without losing trainability and obtain their embeddings.

B. NEURAL PROMPTING WITH TRAINABLE INSTANCE SELECTION

We propose a neural prompting that creates soft prompts for the text generation model from the selection weights via differentiable k-nearest instance selection while the general retriever creates prompts as text. We compose soft prompts from instances softly selected by weighted summing of the embeddings over the instances using the selection weights in Section III-A instead of text prompts. The retriever becomes differentiable by composing with this process.

For the detailed procedure, the k-th selected instance embedding S_k is computed in the retrieval process. Now, we need to prepare soft prompts from selected instances by reshaping selected instance embeddings S to the shape for a soft prompt $P \in \mathbb{R}^{KL \times N}$ by stacking them as $P = [S_{1,1}, S_{1,2}, \ldots S_{1,L}, S_{2,1}, \ldots S_{K,L}]$. Connecting the prompt and a text generation model involves joining the prompt to the input sequence embeddings. In this case, the length of the prompt KL is added to the length of the input series $|x|$, resulting in a new series with the length of $|x| + KL$.

TABLE 1. Statistics of the TACRED Dataset with Different Proportion.

Proportion	Train	Dev.	Test
100%	68,125	22,631	15,509
10%	6,815	2,265	
5%	3,407	1,133	
1%	682	227	

C. TRAINING

The training of the retriever proposed in this paper is simply a matter of optimizing the ETRAG model with the objective function of the target task. Since the retriever consists entirely of differentiable operations in ETRAG, the model with the retriever is end-to-end trainable. Our method innovatively transforms the retriever into an end-to-end trainable model, enhancing its applicability to relation extractors.

The primary challenge in training the retriever is its computational intensity, which requires calculating distances for all instances in D. To mitigate this during training, we employ a strategy of random sampling a subset of instances. This approach significantly reduces the computational burden while maintaining the retriever's efficacy.

Since the base model, SuRE, is subsequently connected to the retriever, a stable training method for the base model is needed. The training process of the text generation model easily affects the retriever's performance. Therefore, we employ a warm-up step in which the retriever is trained in advance. The base model is frozen, and the retriever is updated during the warm-up steps. All the parameters are updated after the warm-up step.

IV. EVALUATION OF RELATION EXTRACTION PERFORMANCE

This section evaluates the relation extraction performance and compares the ETRAG integrated model to existing relation extraction methods. Section IV-A presents the settings of subsequent experiments about datasets, baseline methods, model settings, and training parameters. Based on the settings, Section IV-B shows the performance evaluations of our proposal by comparing it with other methods. Section IV-C is the ablation study to show the elements that affect performance.

A. EXPERIMENTAL SETTINGS

To assess the effectiveness of our relation extraction method, we experiment on the TACRED dataset [55], a standard benchmark in the sentence-level relation extraction, where each instance has an entity pair within a sentence and a gold relation between the pair. TACRED is the only dataset for which templates are available, as it is the dataset of the target evaluated by the base model of SuRE. To understand the impact of training data size, we experimented with reduced training and development data scenarios for TACRED: 100%, 10%, 5%, and 1% of the entire dataset, following existing study [56]. The statistics for each scenario are shown in Tables 1 and 2. We evaluated the mean value of the micro-averaged F1 score for three runs, treating the `no_relation` class as a negative example for TACRED following the official evaluation settings. We calculate the loss to development data for every 100 update steps and report the score obtained when training in early stopping with the early stopping patience of 3 for the TACRED dataset.

> The head entity is ${head entity}. The tail entity is ${tail entity}. The type of ${head entity} is ${label of head entity} . The type of ${tail entity} is ${label of tail entity} . ${input sentence} .

FIGURE 2. The input template of SuRE. ${·} are placeholders to replace with input.

We compared our method against state-of-the-art models: SuRE (based on Pegasus-large) [27], DeepStruct [17], kNN-RE [28], and NLI_DeBERTa [56]. SuRE is, as described in Section II-C, the model extracting relation with the text generation model by formulating the relation extraction task to the text summarization task using Seq2Seq LM. DeepStruct is a PLM trained for structured prediction tasks, which include relation extraction. kNN-RE performs kNN algorithm on the PLM embedding space of relation instances. NLI_DeBERTa extracts relations with a natural language inference task, which recognizes fact inclusion in a hypothesis, by identifying the implication of verbalized relations in a target text.

We employed SuRE [27] as a generation-based relation extraction model. We introduced the ETRAG to SuRE by adding soft prompts into the input sequence between the BOS token and the following prompt. The hyperparameters of the SuRE were the same as those of the original research. The templates were the same as in the original paper: the templates for the SuRE input were in the form of Figure 2, from which the summary templates defined for each relation label are predicted as in Table 3. The beam search width, a parameter used in SuRE classification, was set to 4, the same value as in the original paper of SuRE.

We use the Flan-T5 large model [20] in the SuRE framework with and without the addition of ETRAG because the tuning before experiments showed the training of Pegasus-large [57] based SuRE with ETRAG was unstable. When we introduced ETRAG into the Pegasus-based model, the model predicted only `no_relation` for the same setting. Since our method used two PLMs, the base relation extraction model and the embedding model for the retriever, the larger models were unacceptable for our computational resources. We conducted trials introducing 10-neighbor instances as prompts, i.e., $K = 10$, pretraining the retriever for 300 steps before end-to-end training as the warm-up step described in Section III-C. Due to computational constraints, the database D was constructed from randomly sampled 5,000 instances in the training dataset if training data has more than 5,000 instances. At the training time, 32 instances are randomly sampled as the subset of D before retrieval. For the embeddings E, we used the entity and relation representations of PLM by averaging their spans, where the PLM input was created by concatenating the template in Figure 2 with the relation template. The entity spans were underlined parts of

TABLE 2. Statistics of Relation Labels for Each Split and Proportion of TACRED dataset.

	100%		10%		5%		1%		
Relation Label	Train	Dev.	Train	Dev.	Train	Dev.	Train	Dev.	Test
no_relation	55,112	17,195	5,513	1,721	2,756	861	552	173	12,184
per:title	2,443	919	244	92	122	46	24	9	500
org:top_members/employees	1,890	534	190	53	95	26	19	5	346
per:employee_of	1,524	375	152	38	76	19	15	4	264
org:alternate_names	808	338	80	34	40	17	8	3	213
org:country_of_headquarters	468	177	47	18	24	9	5	2	108
per:countries_of_residence	445	226	44	22	22	11	4	2	148
per:age	390	243	40	24	20	12	4	2	200
org:city_of_headquarters	382	109	38	10	19	5	4	1	82
per:cities_of_residence	374	179	38	18	19	9	4	2	189
per:stateorprovinces_of_residence	331	72	33	8	16	4	3	1	81
per:origin	325	210	32	21	16	11	3	2	132
org:subsidiaries	296	113	30	12	15	6	3	1	44
org:parents	286	96	28	10	14	5	3	1	62
per:spouse	258	159	26	16	13	8	3	2	66
org:stateorprovince_of_headquarters	229	70	23	7	11	4	2	1	51
per:children	211	99	21	10	10	5	2	1	37
per:other_family	179	80	18	8	9	4	2	1	60
org:members	170	85	17	8	9	4	2	1	31
per:siblings	165	30	17	3	9	1	2	0	55
per:parents	152	56	16	6	8	3	2	1	88
per:schools_attended	149	50	14	5	7	3	1	1	30
per:date_of_death	134	206	13	20	6	10	1	2	54
org:founded_by	124	76	12	8	6	4	1	1	68
org:member_of	122	31	12	3	6	1	1	0	18
per:cause_of_death	117	168	12	17	6	9	1	2	52
org:website	111	86	11	9	5	5	1	1	26
org:political/religious_affiliation	105	10	10	1	5	0	1	0	10
per:alternate_names	104	38	10	4	5	2	1	0	11
org:founded	91	38	10	4	5	2	1	0	37
per:city_of_death	81	118	8	12	4	6	1	1	28
org:shareholders	76	55	8	6	4	3	1	1	13
org:number_of_employees/members	75	27	8	2	4	1	1	0	19
per:charges	72	105	8	10	4	5	1	1	103
per:city_of_birth	65	33	7	3	4	1	1	0	5
per:date_of_birth	63	31	7	3	4	1	1	0	9
per:religion	53	53	6	6	3	3	1	1	47
per:stateorprovince_of_death	49	41	4	4	2	2	0	0	14
per:stateorprovince_of_birth	38	26	4	2	2	1	0	0	8
per:country_of_birth	28	20	2	2	1	1	0	0	5
org:dissolved	23	8	2	0	1	0	0	0	2
per:country_of_death	6	46	0	5	0	3	0	1	9

Figure 2. The relation span was the relation template part when the database was embedded and the EOS token when the prediction target was embedded. We applied LoRA [36] to Flan-T5 with rank $r = 32$ and dropout rate 0.1 to all kinds of layers of Transformer. The LoRA is a method for efficient fine-tuning, and we employed it because our pilot experiments showed full tuning and LoRA tuning performances are not significantly different. The dropout rate was set to 0.1. AdamW was used to optimize the models, with a learning rate of 5×10^{-4} for Flan-T5 and 1×10^{-3} for the other parameters, and weights of 5×10^{-6} for the bias and layer normalization parameters. The batch size was set to 64. The parameters used for evaluation were those used when early stopping completed training with patience set to 5. A single NVIDIA A100 was used for each experiment.

B. EXTRACTION PERFORMANCE COMPARISON

The results in Table 4 indicate that ETRAG consistently improved performance from the model without ETRAG for the TACRED dataset. The results confirm that ETRAG can enhance relations extraction by text generation. Additionally, ETRAG outperforms the existing models, SuRE and NLI_DeBERTa, in scenarios with limited training data (10%). This is the new state-of-the-art result for the setting of the TACRED dataset under the 10% setting.

Comparing Pegasus-based and T5-based SuRE, the Pegasus-based SuRE performed better. This is because Pegasus is a model created specifically for the summarization task, which matches SuRE's objective. Even in this situation, the ETRAG boosted the performance of the Flan-T5-based model and achieved the best performance on the 10% setting.

Compared to another instance-based method, kNN-RE, adding generation-based prediction of relations to neighborhood method-based inference confirms the improved extraction performance. This proves that simple instance utilization is insufficient and that inference capability is essential.

TABLE 3. The output template of SuRE. ${·} are placeholder replaced with input.

Relation Label	Template
no_relation	\${subj} has no known relations to \${obj}
per:title	\${subj} is a \${obj}
org:top_members/employees	\${subj} has the high level member \${obj}
per:employee_of	\${subj} is the employee of \${obj}
org:alternate_names	\${subj} is also known as \${obj}
org:country_of_headquarters	\${subj} has a headquarter in the country \${obj}
per:countries_of_residence	\${subj} lives in the country \${obj}
per:age	\${subj} has the age \${obj}
org:city_of_headquarters	\${subj} has a headquarter in the city \${obj}
per:cities_of_residence	\${subj} lives in the city \${obj}
per:stateorprovinces_of_residence	\${subj} lives in the state or province \${obj}
per:origin	\${subj} has the nationality \${obj}
org:subsidiaries	\${subj} owns \${obj}
org:parents	\${subj} has the parent company \${obj}
per:spouse	\${subj} is the spouse of \${obj}
org:stateorprovince_of_headquarters	\${subj} has a headquarter in the state or province \${obj}
per:children	\${subj} is the parent of \${obj}
per:other_family	\${subj} is the other family member of \${obj}
org:members	\${subj} has the member \${obj}
per:siblings	\${subj} is the siblings of \${obj}
per:parents	\${subj} has the parent \${obj}
per:schools_attended	\${subj} studied in \${obj}
per:date_of_death	\${subj} died in the date \${obj}
org:founded_by	\${subj} was founded by \${obj}
org:member_of	\${subj} is the member of \${obj}
per:cause_of_death	\${subj} died because of \${obj}
org:website	\${subj} has the website \${obj}
org:political/religious_affiliation	\${subj} has political affiliation with \${obj}
per:alternate_names	\${subj} has the alternate name \${obj}
org:founded	\${subj} was founded in \${obj}
per:city_of_death	\${subj} died in the city \${obj}
org:shareholders	\${subj} has shares hold in \${obj}
org:number_of_employees/members	\${subj} has the number of employees \${obj}
per:charges	\${subj} is convicted of \${obj}
per:city_of_birth	\${subj} was born in the city \${obj}
per:date_of_birth	\${subj} has birthday on \${obj}
per:religion	\${subj} has the religion \${obj}
per:stateorprovince_of_death	\${subj} died in the state or province \${obj}
per:stateorprovince_of_birth	\${subj} was born in the state or province \${obj}
per:country_of_birth	\${subj} was born in the country \${obj}
org:dissolved	\${subj} dissolved in \${obj}
per:country_of_death	\${subj} died in the country \${obj}

TABLE 4. Comparison of Relation Extraction Performance [%]

	100%	10%	5%	1%
DeepStruct [17]	76.8	–	–	–
SuRE (Pegasus) [27]	75.1	70.7	64.9	52.0
NLI_DeBERTa [56]	73.9	67.9	69.0	63.0
kNN-RE [28]	70.6	–	–	–
SuRE (Flan-T5)	71.4 ±1.6	68.5 ±1.5	65.0 ±3.1	53.5 ±1.4
+ ETRAG	73.3 ±0.5	71.5 ±0.5	68.3 ±2.0	54.6 ±1.1

C. ABLATION STUDIES

This section delves into the factors influencing the extraction performance observed in Section IV-B and investigates the model's behavior. We conducted ablation studies in the 10% training instance setting for the TACRED dataset to confirm when our method showed notable improvements.

Our ablation study aimed to pinpoint the elements critical to our method's enhanced performance. We examined various scenarios: employing k-nearest neighbor instances without retriever training (No Retriv. Training), omitting the warm-up phase in Retriever training (No Warm-up), using randomly chosen instances (Random), and utilizing CLS token representations (CLS).No Retriv. Training aims to investigate the usefulness of the end-to-end trainable retriever by using the initial parameter for the retriever and operations of ETRAG. No Warm-up omits the warm-up step but trains the retriever, which checks the stability effect of the warm-up. Random picks up instances randomly and uses soft prompts in the same embedding process as ETRAG, where the experiment checks retrieval process training effectiveness. CLS uses only CLS token representations instead of the relation extraction-specific representation. The other settings of experiments are the same as settings in Section IV-A except for the number of runs for evaluation that changed from 3 runs to 1 run.

The results in Table 5 reveal that omitting any of these components results in lower F1 scores, underscoring their collective importance. The No Retriv. Training caused a performance loss of 2.2 percentage points, which was not much different from the performance of SuRE without any retrievers. This result indicates that in the relation extraction

TABLE 5. Ablation Study Results [%]

SuRE (Flan-T5)	68.5
+ ETRAG	71.7
No Retriv. Training	69.2
No Warm-up	70.7
Random	71.0
CLS	68.8

with text generation model, the retriever needs to be trained for the relation extraction objective to improve performance when relation instances are introduced with the retriever.

The No Warm-up reduced extraction performance by 1.0 percentage points; the decrease was relatively smaller than in the other cases, No Retriv. Training and CLS. This may be due to the lack of treatment for convergence stability, although similar processing and training of the model is carried out.

In the case of Random, where random instances are used without training the retrieval process, the performance drop is relatively small, around 0.7 percentage points. This comparison allows us to evaluate the improvement separately due to the introduction and selection of instances. Compared to the case where no retrieval process is used (i.e., SuRE), introducing randomly selected instances shows a 2.4 percentage point improvement. The results suggest that introducing examples improves performance, and further progress is made when the model selects instances.

For comparison of representations used in ETRAG, the performance with the CLS representation degraded 2.9 percentage points from the one engineered for relation extraction. Additionally, the performance was almost the same as one of SuRE without a retriever. Thus, engineering a representation specializing in the target task is essential to using ETRAG.

Overall, these analyses highlight the contributions of our method, ETRAG, to performance, and we confirmed that all its components are effective. Instance selection and training helped improve relation extraction outcomes.

V. RETRIEVER ANALYSIS

Section IV showed the performance improvement by our proposed ETRAG that was integrated into the text generation model (Section IV-B). The ablation studies in Section IV-C showed the factors contributing to performance. However, it is not yet clear what happened inside ETRAG that led to the improvements. Therefore, we also analyzed the retriever from the perspective of instances. Section V-A confirms the impact of the instances by changing the number of instances created by the retriever and checking their behavior at that time; Section V-B shows what instances were retrieved and used after training by statistically analyzing the instances to analyze the actual retrieved instances directly.

A. PERFORMANCE VARIATION WITH NUMBER OF INSTANCES

We investigated the sensitivity of the number of instances k by varying k from 0 to 20, where 0 instance means ETRAG is not used. The F1, precision, and recall scores versus k are shown in Figure 3. The F1 score reached its maximum at $k = 10$ and no significant change in the $1 \leq k \leq 15$ interval. On the other hand, the balance between precision and recall changed significantly. In the $0 \leq k \leq 5$ interval, precision increased, and recall decreased as overall performance improved. Conversely, recall gradually increased, and precision decreased in the interval $k > 5$, except for $k = 20$. Since precision and recall were balanced at $k = 10$, the F1 score was the largest, defined as the harmonic mean of the precision and recall.

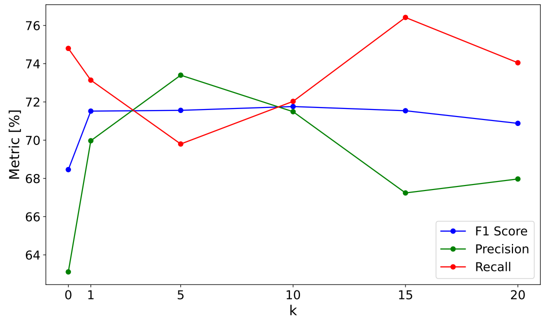


FIGURE 3. Change in Extraction Performance with the Number of Instances *k* Used in Prompts

TABLE 6. Comparison of contents between retrieved instances and extraction target. Label means the relation label is the same. Entity means either head entity or tail entity is contained. Related means either head entity, tail entity, or relation label is contained.

	Label	Head entity	Tail entity	Entity	Related
top-1	72.2	7.1	3.4	10.0	73.6
top-3	67.9	8.3	4.4	12.0	77.4
top-5	61.8	10.9	6.3	15.7	78.9
top-10	60.4	18.6	10.9	25.8	82.1

These characteristics are useful in real-world applications and can be used to make performance trade-offs of precision and recall to suit the situation. For example, applications that require users to retrieve necessary relational information, such as a search system, could use a larger k for coverage.

B. RETRIEVED INSTANCES

Since the characteristics of retrievers are most evident in retrieved instances, we investigate the retrieved instances in ETRAG. However, because selected instances in ETRAG are a weighted sum of the instance embeddings, obtaining what was chosen explicitly is impossible. Therefore, we analyze k nearest instances specified by the actual kNN algorithm on the feature space after training of ETRAG, which turns out that they are almost the same instances used in ETRAG.

We took statistics on the retrieved instances linked to the extraction target. The targets for statistics are the relation labels, head entities, and tail entities, which are closely related to the relation extraction. We calculate the percentage of matches between the retrieved instances and the target of extraction in relation labels and the surface of entities.

The statistics in Table 6 show the percentage of the retrieved instances that contain objects related to relation extraction. First, for the Label column, the percentage gradually decreases from top-1 to top-10. This indicates that the feature space of the retrieval is structured based on the type of relation labels and that instances with the same relation labels are placed close to each other. For the Entity column, the percentage gradually increases from top-1 to top-10, indicating that including entities is a second retrieval perspective, although the percentages are less than those in the Label column. The results for the Head and Tail entity columns show that instances containing the head entity are more intensive. From the results of the top-10 row in the Related column, more than 80% have been selected that contain labels or entities relevant for relation extraction. This may be due to the use of relation and entity features for retrieval. As a result of end-to-end training from these features, the distance becomes smaller when the relation labels match or entities are included. These properties were not given intentionally but were acquired only by training through end-to-end relation extraction, indicating that the retriever does not require any special training.

VI. CONCLUSIONS

This study introduced a novel approach to the text generation model by implementing a retriever with the differentiable k-nearest neighbor selection for end-to-end trainable modeling. Existing models with retrievers cannot train end-to-end due to the non-differentiable environments of the instance selection part and the integration part of instances. Therefore, we proposed a fully differentiable and end-to-end trainable RAG ETRAG by differentiable k-nearest neighbor selection and integration as a soft prompt. Our method, centered around neural prompting, significantly enhances the retriever's capability to select instances for use in prompts.

Experimental findings underscore this approach's effectiveness, particularly in scenarios with limited training data in evaluating relation extraction performance. We evaluated the model with ETRAG and compared the model without ETRAG with existing methods with the TACRED dataset. Our experiments showed that our proposal ETRAG consistently improved from the baseline model without ETRAG. Moreover, the model reported outstanding performance in low-resource settings, especially the new state-of-the-art for the TACRED dataset in the 10% training data setting.

Our analysis confirmed that the number of retrieved instances introduced by ETRAG can balance the precision-recall trade-off. We also confirmed that the end-to-end trained retriever referred to the instances involved in relation extraction. However, our study also identified limitations in our method's performance when training instances are sufficient.

Future work could focus on refining the retriever's training process to adapt more effectively to varying sizes of training datasets and exploring ways to optimize instance selection for a broader range of data scenarios. Moreover, since we evaluated on only relation extraction while ETRAG can be applied to other text generation models, further evaluation of other tasks, such as question answering [58], will be conducted to confirm the applicability of ETRAG.

ACKNOWLEDGEMENTS

The authors would like to express gratitude to the members of the Computational Intelligence Laboratory for their helpful comments on this study. This work was supported by Grant-in-Aid for JSPS Fellows Grant Number JP22J14025. Computational resources of AI Bridging Cloud Infrastructure (ABCI) provided by the National Institute of Advanced Industrial Science and Technology (AIST) were used.

REFERENCES

[1] N. A. Chinchor, "Overview of MUC-7," in *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, 1998. [Online]. Available: https://aclanthology.org/M98-1001

[2] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003, pp. 142–147. [Online]. Available: https://www.aclweb.org/anthology/W03-0419

[3] L. Wang, Z. Cao, G. de Melo, and Z. Liu, "Relation classification via multi-level attention CNNs," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1298–1307. [Online]. Available: https://aclanthology.org/P16-1123

[4] K. Du, B. Yang, S. Wang, Y. Chang, S. Li, and G. Yi, "Relation extraction for manufacturing knowledge graphs based on feature fusion of attention mechanism and graph convolution network," *Knowledge-Based Systems*, vol. 255, p. 109703, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122008620

[5] A. P. Davis, C. G. Murphy, C. A. Saraceni-Richards, M. C. Rosenstein, T. C. Wiegers, and C. J. Mattingly, "Comparative Toxicogenomics Database: a knowledgebase and discovery tool for chemical–gene–disease networks," *Nucleic Acids Research*, vol. 37, no. suppl_1, pp. D786–D792, 09 2008. [Online]. Available: https://doi.org/10.1093/nar/gkn580

[6] K. Zhu, J. Huang, and K. C.-C. Chang, "Descriptive knowledge graph in biomedical domain," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Y. Feng and E. Lefever, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 462–470. [Online]. Available: https://aclanthology.org/2023.emnlp-demo.42

[7] Y. Long, M. Wu, Y. Liu, Y. Fang, C. K. Kwoh, J. Chen, J. Luo, and X. Li, "Pre-training graph neural networks for link prediction in biomedical networks," *Bioinformatics*, vol. 38, no. 8, pp. 2254–2262, 02 2022. [Online]. Available: https://doi.org/10.1093/bioinformatics/btac100

[8] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMs on sequences and tree structures," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1105–1116. [Online]. Available: https://www.aclweb.org/anthology/P16-1105

[9] G. Nan, Z. Guo, I. Sekulic, and W. Lu, "Reasoning with latent structure refinement for document-level relation extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 1546–1557. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.141

[10] Y. Ma, A. Wang, and N. Okazaki, "DREEAM: Guiding attention with evidence for improving document-level relation extraction," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, A. Vlachos and I. Augenstein, Eds. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 1971–1983. [Online]. Available: https://aclanthology.org/2023.eacl-main.145

[11] E. Riloff, "Automatically constructing a dictionary for information extraction tasks," in *AAAI*, vol. 1, no. 1. Citeseer, 1993, pp. 2–1.

[12] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1083–1106, 2003.

[13] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter*

of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. [Online]. Available: https://www.aclweb.org/anthology/N18-1202

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[16] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[17] C. Wang, X. Liu, Z. Chen, H. Hong, J. Tang, and D. Song, "DeepStruct: Pretraining of language models for structure prediction," in *Findings of the Association for Computational Linguistics: ACL 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 803–823. [Online]. Available: https://aclanthology.org/2022.findings-acl.67

[18] J. Y. Huang, B. Li, J. Xu, and M. Chen, "Unified semantic typing with meaningful label inference," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 2642–2654. [Online]. Available: https://aclanthology.org/2022.naacl-main.190

[19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[20] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling instruction-finetuned language models," 2022.

[21] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023.

[22] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, jan 2020.

[23] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[24] A. D. Cohen, S. Rosenman, and Y. Goldberg, "Supervised relation classification as twoway span-prediction," in *4th Conference on Automated Knowledge Base Construction*, 2022.

[25] S. Wadhwa, S. Amir, and B. Wallace, "Revisiting relation extraction in the era of large language models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 15 566–15 589. [Online]. Available: https://aclanthology.org/2023.acl-long.868

[26] Z. Wan, F. Cheng, Z. Mao, Q. Liu, H. Song, J. Li, and S. Kurohashi, "GPT-RE: In-context learning for relation extraction using large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 3534–3547. [Online]. Available: https://aclanthology.org/2023.emnlp-main.214

[27] K. Lu, I.-H. Hsu, W. Zhou, M. D. Ma, and M. Chen, "Summarization as indirect supervision for relation extraction," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 6575–6594. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.490

[28] Z. Wan, Q. Liu, Z. Mao, F. Cheng, S. Kurohashi, and J. Li, "Rescue implicit and long-tail cases: Nearest neighbor relation extraction," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 1731–1738. [Online]. Available: https://aclanthology.org/2022.emnlp-main.113

[29] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[32] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3045–3059. [Online]. Available: https://aclanthology.org/2021.emnlp-main.243

[33] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. [Online]. Available: https://aclanthology.org/2021.acl-long.353

[34] Y. Tian, H. Song, Z. Wang, H. Wang, Z. Hu, F. Wang, N. V. Chawla, and P. Xu, "Graph neural prompting with large language models," 2023.

[35] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 2790–2799. [Online]. Available: https://proceedings.mlr.press/v97/houlsby19a.html

[36] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=nZeVKeeFYf9

[37] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *Journal of the American Statistical Association*, vol. 53, no. 282, pp. 457–481, 1958. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1958.10501452

[38] A. Søgaard, "Semi-supervised condensed nearest neighbor for part-of-speech tagging," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, D. Lin, Y. Matsumoto, and R. Mihalcea, Eds. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 48–52. [Online]. Available: https://aclanthology.org/P11-2009

[39] Y. Yang and A. Katiyar, "Simple and effective few-shot named entity recognition with structured nearest neighbor learning," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 6365–6375. [Online]. Available: https://aclanthology.org/2020.emnlp-main.516

[40] H. Ouchi, J. Suzuki, S. Kobayashi, S. Yokoi, T. Kuribayashi, M. Yoshikawa, and K. Inui, "Instance-based neural dependency parsing," *Transactions of the Association for Computational*

Linguistics, vol. 9, pp. 1493–1507, 2021. [Online]. Available: https://aclanthology.org/2021.tacl-1.89

[41] I. Mani and I. Zhang, "knn approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of workshop on learning from imbalanced datasets*, vol. 126, no. 1. ICML, 2003, pp. 1–7.

[42] C. Walker and L. D. Consortium, *ACE 2005 Multilingual Training Corpus*, ser. LDC corpora. Linguistic Data Consortium, 2005. [Online]. Available: https://books.google.co.jp/books?id=SbjjuQEACAAJ

[43] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz, "SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala, Sweden: Association for Computational Linguistics, Jul. 2010. [Online]. Available: https://www.aclweb.org/anthology/S10-1006

[44] S. K. Sahu, F. Christopoulou, M. Miwa, and S. Ananiadou, "Inter-sentence relation extraction with document-level graph convolutional neural network," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 4309–4316. [Online]. Available: https://www.aclweb.org/anthology/P19-1423

[45] F. Christopoulou, M. Miwa, and S. Ananiadou, "Connecting the dots: Document-level neural relation extraction with edge-oriented graphs," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4925–4936. [Online]. Available: https://www.aclweb.org/anthology/D19-1498

[46] M. Miwa and Y. Sasaki, "Modeling joint entity and relation extraction with table representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1858–1869. [Online]. Available: https://www.aclweb.org/anthology/D14-1200

[47] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, J. Tsujii and J. Hajic, Eds. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 2335–2344. [Online]. Available: https://aclanthology.org/C14-1220

[48] W. Zhou, K. Huang, T. Ma, and J. Huang, "Document-level relation extraction with adaptive thresholding and localized context pooling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[49] T.-J. Fu, P.-H. Li, and W.-Y. Ma, "GraphRel: Modeling text as relational graphs for joint entity and relation extraction," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1409–1418. [Online]. Available: https://www.aclweb.org/anthology/P19-1136

[50] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, and M. Sun, "FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 4803–4809. [Online]. Available: https://aclanthology.org/D18-1514

[51] Z. Wu, Y. Wang, J. Ye, Z. Wu, J. Feng, J. Xu, and Y. Qiao, "OpenICL: An open-source framework for in-context learning," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, D. Bollegala, R. Huang, and A. Ritter, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 489–498. [Online]. Available: https://aclanthology.org/2023.acl-demo.47

[52] T. Plötz and S. Roth, "Neural nearest neighbors networks," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[53] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=rkE3y85ee

[54] P. Verga, E. Strubell, and A. McCallum, "Simultaneously self-attending to all mentions for full-abstract biological relation extraction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 872–884. [Online]. Available: https://www.aclweb.org/anthology/N18-1080

[55] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 35–45. [Online]. Available: https://aclanthology.org/D17-1004

[56] O. Sainz, O. Lopez de Lacalle, G. Labaka, A. Barrena, and E. Agirre, "Label verbalization and entailment for effective zero and few-shot relation extraction," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1199–1212. [Online]. Available: https://aclanthology.org/2021.emnlp-main.92

[57] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 11 328–11 339. [Online]. Available: https://proceedings.mlr.press/v119/zhang20ae.html

[58] G. Kim, S. Kim, B. Jeon, J. Park, and J. Kang, "Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 996–1009. [Online]. Available: https://aclanthology.org/2023.emnlp-main.63

KOHEI MAKINO received B.E. and M.E. degrees from Toyota Technological Institute, Aichi, Nagoya, Japan, where he is currently pursuing a doctor's degree. His research interests include deep learning, natural language processing, and information extraction.

MAKOTO MIWA received a Ph.D. degree from the University of Tokyo in 2008. He is currently a professor at the Toyota Technological Institute and an invited researcher at the National Institute of Advanced Industrial Science and Technology. His research interests include natural language processing, deep learning, and information extraction.

YUTAKA SASAKI received his B.E., M.Eng. and Ph.D. in Engineering from the University of Tsukuba in 1986, 1988, 2000, respectively. From 1988 to 2006, he was with the NTT laboratories, Japan. During 2004-2006, he was a department head at the Advanced Telecommunication Research Institute International (ATR), Kyoto, Japan. In 2006, he joined the School of Computer Science, the University of Manchester and the National Center for Text Mining (NaCTeM) in UK. In 2009, he moved to the Toyota Technological Institute, Nagoya, Japan since then he is a Professor at the Department of Advanced Science and Technology. He is also an adjoint professor at the Toyota Technological Institute at Chicago. His recent research interests include machine learning, natural language processing, deep state-space models, and biomedical/materials informatics.