Wiki-based Prompts for Enhancing Relation Extraction using Language Models

Amirhossein Layegh
KTH Royal Institute of Technology
Stockholm, Sweden
amlk@kth.se

Amir H. Payberah
KTH Royal Institute of Technology
Stockholm, Sweden
payberah@kth.se

Ahmet Soylu
Oslo Metropolitan University
Oslo, Norway
ahmet.soylu@oslomet.no

Dumitru Roman
SINTEF AS
Oslo, Norway
dumitru.roman@sintef.no

Mihhail Matskin
KTH Royal Institute of Technology
Stockholm, Sweden
misha@kth.se

ABSTRACT

Prompt-tuning and instruction-tuning of language models have exhibited significant results in few-shot Natural Language Processing (NLP) tasks, such as Relation Extraction (RE), which involves identifying relationships between entities within a sentence. However, the effectiveness of these methods relies heavily on the design of the prompts. A compelling question is whether incorporating external knowledge can enhance the language model's understanding of NLP tasks. In this paper, we introduce *wiki-based* prompt construction that leverages Wikidata as a source of information to craft more informative prompts for both prompt-tuning and instruction-tuning of language models in RE. Our experiments show that using wiki-based prompts enhances cutting-edge language models in RE, emphasizing their potential for improving RE tasks. Our code and datasets are available at GitHub [1].

CCS CONCEPTS

• **Computing methodologies** → Information extraction; **Natural language processing**.

KEYWORDS

Relation Extraction, Language Models, Prompt Construction, knowledge Integration

ACM Reference Format:
Amirhossein Layegh, Amir H. Payberah, Ahmet Soylu, Dumitru Roman, and Mihhail Matskin. 2024. Wiki-based Prompts for Enhancing Relation Extraction using Language Models. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3605098.3635949

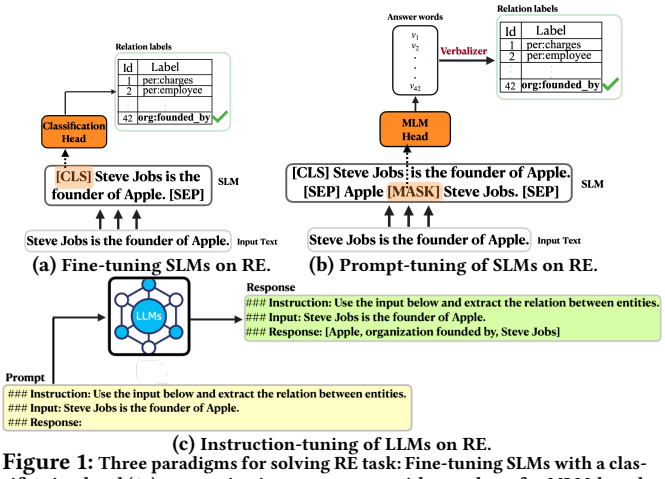

Figure 1: Three paradigms for solving RE task: Fine-tuning SLMs with a classification head (1a), prompting input sentences with templates for MLM-based prediction (1b), and instruction-tuning LLMs to generate relation-containing responses (1c).

1 INTRODUCTION

Relation Extraction (RE) is a fundamental task in Natural Language Processing (NLP), identifying and categorizing semantic relationships between *entities* mentioned in the text. RE is important in many NLP tasks such as information extraction, knowledge base construction, knowledge graph creation, and question answering by enabling the extraction of structured information from unstructured textual data [3, 27, 28].

Most prior research on RE focuses on adapting Standard-scale Language Models (SLMs) such as BERT [9] to downstream RE tasks [18, 45]. In this paradigm, we fine-tune SLMs on RE tasks, utilizing a classification head to predict the relation between entities (Figure 1a). Although this approach is practical, it involves challenges such as being time-consuming, requiring lots of annotated data, and lack of generalization, especially in few-shot RE. One method to overcome these limitations is to prompt-tune SLMs by reframing the RE task as a Masked Language Modeling (MLM) problem. This reframing is achieved by employing a textual prompt template to fill a blank in a given prompt, predicting the relation between entities (Figure 1b) [6, 11, 13]. The predicted blank is then

[1] https://github.com/AmirLayegh/Wiki-basedRE

The work in this paper was partially funded by the projects DataCloud (H2020 101016835), enRichMyData (HE 101070284), Graph-Massivizer (HE 101093202), UPCAST (HE 101093216), and BigDataMine (NFR 309691). The experiments were enabled by Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

SAC '24, April 8–12, 2024, Avila, Spain

ACM ISBN 979-8-4007-0243-3/24/04.
https://doi.org/10.1145/3605098.3635949

linked to actual relation labels using a verbalizer [35]. Although prompt-tuning shows impressive results in few-shot RE, model performance heavily relies on costly prompt and verbalizer engineering to discover the optimal prompt template and answer space for RE [17, 25].

Recently, there has been a significant increase in model sizes with Large-scale Language Models (LLMs) such as GPT-3 [5], and Llama 2 [39] each containing billions of parameters. Unlike prompt-tuning approaches, these generative models can be applied directly to tasks without explicit verbalization. However, a significant challenge with LLMs is their focus on word prediction within context, which may not align with the user's desire to understand and follow their instructions [50]. To address this, Supervised Fine-Tuning (SFT) of LLMs known as *instruction-tuning* has been proposed, involving fine-tuning LLMs on datasets containing human-written instructions and context to align the model's behavior more closely with the user's expectations [31]. Figure 1c illustrates the process of instruction-tuning LLMs for RE tasks. This approach generates responses that encapsulate the relation between entities within the given input sentence by utilizing the task instruction input sentence as a prompt.

Despite the considerable success of prompt-tuned SLMs and instruction-tuned LLMs across various applications and scenarios, including standard and few-shot settings, they have been denounced for memorizing facts and knowledge in the training corpus [15]. This issue becomes particularly pronounced in semantically complex tasks such as RE, requiring domain-specific knowledge and expertise for generalization. To address these limitations and further enhance the effectiveness of RE models, we propose a novel methodology that leverages external knowledge sources, particularly Wikidata [2], to construct informative prompts for RE tasks. We refer to these prompts as *wiki-based* prompts, aiming to provide additional context and information to assist the model in understanding and extracting relations between entities in text.

In this paper, we introduce the detailed methodology for constructing wiki-based prompts, integrating them into the prompt-tuning process of SLMs, and exploring their effectiveness in the instruction-tuning of LLMs. In summary, we present the following contributions:

- We propose wiki-based prompts, a novel approach for enhancing RE tasks, leveraging external knowledge from Wikidata to create informative prompts.
- We introduce a methodology for prompt-tuning of SLMs using these wiki-based prompts, addressing the challenge of efficient prompt template construction.
- We extend the exploration to instruction-tuned LLMs and demonstrate the application of wiki-based prompts combined with SFT techniques to align LLMs more closely with human instruction for RE tasks.
- We employ advanced SFT strategies, including Low-Rank Adaptation (LoRA) SFT [16] and Direct Preference Optimization (DPO) [32], to enhance the performance of LLMs in RE tasks, particularly in the context of few-shot RE.
- We conduct comprehensive experiments and evaluations on four publicly available RE datasets to assess the effectiveness of our wiki-based prompts and the impact of instruction-tuning and SFT techniques on RE tasks, showcasing improved generalization and performance.

The paper is structured as follows: Section 2 introduces RE using SLMs and LLMs. In Section 3, we present our wiki-based prompt construction, detailing its incorporation into prompt-tuning for SLMs and instruction-tuning for LLMs. Section 4 covers experimental details, results, dataset information, and evaluation metrics. Section 5 overviews related work in RE and language models. Finally, Section 6 summarizes contributions, discusses findings, and suggests future research directions.

2 BACKGROUND

Relation extraction (RE) aims to identify and classify the relationship between a *subject* and an *object* entities mentioned in a sentence. In an RE dataset, an example typically is a pair of $(\mathbf{X}_i, \mathbf{Y}_i)$, where $\mathbf{X}_i = \{x_1, x_2, \cdots, s, \cdots, o, \cdots, x_n\}$, is an input sentence containing n tokens, and s and o indicate subject and object entities, respectively. $\mathbf{Y}_i \in \mathcal{Y}$ is the corresponding relation label showing the *relationship* between s and o, and $\mathcal{Y}$ is a set of pre-defined relation labels such as `org:founded`, `per:charges`, and `org:subsidiary`. A popular approach for RE tasks is to use language models. In this section, we review two types of language models for the RE tasks: Standard-scale Language Models (SLMs), such as BERT [9] and RoBERTa [26], and Large-scale Language Models (LLMs), such as Llama 2 [39] and GPT-3 [5].

2.1 SLMs for RE

Fine-tuning SLMs on downstream RE tasks is a common approach to training a model for RE tasks [18, 45, 47, 51]. In this approach, an SLM $\mathcal{S}$, pre-trained on massive unlabeled text data, is fine-tuned on a labeled RE dataset. During the fine-tuning step, each input sentence is converted into a sequence of tokens with a special classification token and an end-of-sequence token. The SLM $\mathcal{S}$ then encodes all sentence tokens into hidden vectors and uses a label-specific classifier to compute the probability distribution of the classification token hidden vector over the relation label space (as illustrated in Figure 1a).

However, fine-tuning SLMs on few-shot RE tasks, where very few examples of each relation label $\mathbf{Y}_i$ are available in the dataset, is challenging. This is mainly due to the gap between pre-training and fine-tuning objectives. Prompt-tuning of SLMs is an approach to bridge this gap by reformulating the downstream RE task as a Masked Language Modeling (MLM) problem using a textual prompt template. This way, the fine-tuning stage becomes more similar to the problem solved during pre-training. To do so, we use a *prompt template* $\mathcal{T}(.)$ to convert an input sentence $\mathbf{X}_i$ to a format suitable for the SLM $\mathcal{S}$ to perform MLM. For example, the prompt template for RE can be $\mathcal{T}(\mathbf{X}_i)$ = `The relation between [subject] and [object] is [MASK].` We also need a *verbalizer* $\mathcal{M}$ to map the predicted word for `[MASK]` to a relation label $\mathbf{Y}_i \in \mathcal{Y}$ (as illustrated in Figure 1b).

Although prompt-tuning of SLMs has shown promising results on RE tasks, particularly on few-shot RE, the effectiveness of the learning process significantly relies on finding the optimal prompt template and verbalizer. This search for an optimal prompt template and verbalizer can hinder this paradigm [17, 24].

[2] https://www.wikidata.org

2.2 LLMs for RE

LLMs, such as GPT-3 [5] and Llama 2 [39], are usually very good at generating grammatically correct and semantically meaningful text. However, despite their outstanding performance, they can produce false information, bias, and toxic text [4]. One approach to address this issue is to prompt LLMs with task-specific solved examples, helping them learn patterns and perform a range of few-shot NLP tasks [5]. Another approach is to fine-tune LLMs using human-written instructions (a.k.a *instruction-tuning*) [8, 29, 31]. Nevertheless, instruction-tuning requires appropriate annotated human-written instruction data. Moreover, fine-tuning LLMs with billions of parameters on instruction data is computationally expensive. In the rest of this section, we explore how to create instruction RE data and efficiently fine-tune LLMs on instruction RE data using Parameter-Efficient Fine-Tuning (PEFT). Furthermore, we investigate the effective alignment of LLMs based on human preference responses employing Direct Preference Optimization (DPO).

2.2.1 ***Instruction-tuning of LLMs for RE.*** By providing LLMs with specific instructions, we can guide them toward producing more accurate and informative text. Each example in the instruction data contains three parts: (1) *Instruction* $\mathbf{I}$, which is a text that describes the RE task, for example, $\mathbf{I}$: `find the relation between the two entities in the sentence`, (2) *Context* or *Input* $\mathbf{X}_i$, which is the context of the RE task, including the input sentence, for example, $\mathbf{X}_i$: `{"Steve Jobs is the founder of Apple."}`, and (3) *Response* or *Output* $\mathbf{Y}_i$, which is an appropriate response for relation label between entities mentioned in the Input $\mathbf{X}_i$, for example, $\mathbf{Y}_i$: `[Apple, founded by, Steve Jobs]` (Figure 1c).

An LLM $\mathcal{L}$ receives a task instruction $\mathbf{I}$ alongside the corresponding context $\mathbf{X}_i$ and produces the response $\mathbf{Y}_i$, i.e., $\mathcal{L}(\mathbf{I}, \mathbf{X}_i) = \mathbf{Y}_i$. The LLM $\mathcal{L}$ then is fine-tunned by tweaking its parameters to reduce the loss function, which is typically the cross-entropy loss between the predicted $\mathbf{Y}_i$ and the ground-truth response. We call this approach of fine-tuning LLMs as *Supervised Fine-Tuning (SFT)*.

2.2.2 ***Fine-tuning of LLMs More Efficiently.*** One approach to reducing the computational cost of fine-tuning LLMs is using Parameter-Efficient Fine-Tuning (PEFT) techniques, such as prefix tuning [21], LLaMA-Adapter [48], and Low-rank adaptation (LoRA) [16]. These techniques reduce the computational cost by only updating a subset of the LLM's parameters. For example, the fundamental idea behind LoRA lies in the ability of LLMs to acquire knowledge from inputs with reduced dimensionality [16].

Another approach to improving the efficiency of fine-tuning LLMs is to use Reinforcement Learning (RL). RL from Human Feedback (RLHF) is a standard final step of SFT of LLMs [31]. It ensures that the LLM's response follows provided instructions and refrains from generating inaccurate information [31]. However, RLHF can be unstable, primarily due to the complexity of hand-crafting effective reward functions while preventing deviations from the original SFT of LLM [32, 43]. *Direct Preference Optimization (DPO)* [32] is a novel training paradigm to align SFT of LLMs from human preferences. This approach eliminates the need to train a reward function by identifying a mapping from the LLM policy to the reward function that maximizes the expected reward of the LLM.

3 METHOD

This section outlines our approach to enhancing RE using our new way of constructing prompts, which we call *wiki-based* prompts, within the context of SLMs and LLMs. Here, we first explain how to construct wiki-based prompts utilizing the Wikidata knowledge graph. Next, we explore how to enhance the prompt-tuning of SLMs by integrating our wiki-based prompts. Finally, we investigate the efficient RE using instruction-tuning LLMs and align them with an RL-based technique, all facilitated by our wiki-based prompts.

3.1 Wiki-based Prompt Construction

In our RE approach, we leverage Wikidata, a comprehensive knowledge graph, to devise our *wiki-based* prompts designed for RE tasks. Wikidata is a structured knowledge base that contains knowledge about entities, their properties, and their relationships. We use this knowledge to create more informative and relevant prompts for RE tasks. By combining techniques such as entity markers and the wealth of knowledge graph information, we aim to elevate the performance of RE models. Below, we discuss our approach to creating such prompts.

3.1.1 ***Entity Markers.*** Inspired by [51], we integrate entity markers, represented by specialized token pairs, into our prompt construction process to explicitly highlight *subject* and *object* entities within the input sentence. We use `[E1]` and `[/E1]` tokens for subjects, and `[E2]` and `[/E2]` tokens for objects. For instance, by transforming the input sentence $\mathbf{X}_i$ = `{"Steve Jobs is the founder of Apple."}` using the subject marker `[E1]` and the object marker `[E2]`, we create the entity marked input sentence $\mathbf{X}_i$ = `{"[E2] Steve Jobs [/E2] is the founder of [E1] Apple [/E1]."}`, highlighting the entities of interest for RE.

3.1.2 ***Wikidata for Prompt Construction.*** To construct the wiki-based prompt, we leverage the extensive knowledge in Wikidata by querying the *instance_of* attribute from Wikidata and integrating this attribute into the prompts. The *instance_of* attribute provides a categorical perspective categorizing entities based on their types. For example, for an entity representing `Steve Jobs`, the *instance_of* attribute can be `person`, which helps to classify the entity as a human being (see Figure 2). We specifically focus on the *instance_of* attribute for its foundational and semantically rich classification, effectively characterizing entities based on their types for relation extraction. This categorical approach is chosen for its interpretability and direct relevance to the RE task despite the wealth of information available in Wikidata.

To ensure clarity and consistency in our categorization process while minimizing the potential for misleading the language model, we have developed a *schema* for the *instance_of* attribute sourced from Wikidata. This schema provides a structured framework for refining the categorical perspective, resulting in more reliable categorization outcomes. For example, within our schema, entities with *instance_of* attribute such as `calendar year`, `decade`, and `aspect of history` are classified as sub-categories of `time` or entities with *instance_of* attribute such as `enterprise`, `business`, and `airline` are classified as sub-categories of `organization`. Since fine-tuning the model on particular entity types can make the model extremely sensitive to variations in data, leading to suboptimal performance when faced with novel or less common entity types, schema-based

classification can help the model remain robust across a broader spectrum of inputs. Moreover, by using a more general category, such as `organization`, we ensure uniformity in the treatment of related entities across various data sources and contexts.

If the *instance_of* attribute cannot be retrieved from Wikidata, we seamlessly substitute it with querying the *entity_description* attribute that provides concise explanations and summaries of the entities, including vital attributes, relationships, and contextual information. For instance, for an item indicating the concept of `Computer`, the *instance_of* attribute is null; thus, we refer to the *entity_description* in Wikidata, `"general-purpose device for performing arithmetic or logical operations"`. This adaptive strategy ensures the high-quality creation of wiki-based prompts for effective RE.

One challenge here is to disambiguate entity mentions to identify the correct referent of entity mentions in a text. For example, the entity mention `Apple` in a sentence could refer to the `Apple fruit` or the `Apple corporation`. To address this issue, we employ *BLINK* [22], a Python library utilizing Wikipedia [3] as a knowledge base. To create the wiki-based prompts, we establish connections between entity mentions and their corresponding Wikidata entities by extracting Wikipedia page titles using BLINK and linking them to the relevant Wikidata items. This association allows us to exploit each entity's rich knowledge graph information. Figure 2 shows detailed examples of the wiki-based prompt construction.

3.2 Prompt-tuning of SLMs Using Wiki-based Prompts

After constructing wiki-based prompts, we use them in the prompt-tuning process of SLMs. As discussed in Section 2.1, this process requires defining (1) the prompt template and (2) the verbalizer. Here, we explain how to create these two components.

3.2.1 ***Prompt Template Creation.*** Developing a prompt template is crucial to achieving excellent performance in the prompt-tuning of SLMs. Applying the prompt template to the input sentence $\mathbf{X}$ yields the prompted input sentence $\mathbf{X}_{prompt}$. Based on the idea of creating wiki-based prompts, discussed in Section 3.1, our wiki-based prompt template, denoted as $\mathcal{T}$, comprises four critical components (see Figure 2):

- The entity-marked input sentence (details in Section 3.1.1).
- The subject entity with its Wikidata knowledge extracted using *instance_of* or *entity_description*.
- The object entity with its Wikidata knowledge extracted using *instance_of* or *entity_description*.
- A `[MASK]` token that enables the SLM $\mathcal{S}$ to perform MLM and predict the appropriate word for the `[MASK]` token as a placeholder of relation label between the entities.

Consider the given input sentences $\mathbf{X}_1$ = {`Steve Jobs is the founder of Apple.`} with subject entity `Apple`, object entity `Steve Jobs`, and relation `org:founded_by` and $\mathbf{X}_2$ = {`Marcus Berg was born in Sweden.`}, with subject entity `Marcus Berg`, object entity `Sweden`, and relation `per:country_of_birth`. Applying wiki-based prompt template $\mathcal{T}(.)$ to these sentences yields the prompted input sentences illustrated in Figure 2. By integrating Wikidata knowledge extracted using *instance_of* or *entity_description*, our prompts infuse additional context and relevant information, enhancing understanding by SLMs and enabling superior generalization, particularly in scenarios with limited resources.

3.2.2 ***Verbalizer Creation.*** We aim to use the predicted word for the `[MASK]` token by the SLM $\mathcal{S}$ to get the relation label between the entities. However, the predicted word for `[MASK]` may not be the same as the actual label; thus, we need a verbalizer to map the predicted word to an actual label. However, in most prompt-tuning approaches, the verbalizer is manually designed by humans [35], making it challenging to develop effective verbalizers for a particular task automatically. It becomes more challenging in RE tasks, where relation labels with rich semantic knowledge (e.g., `per:place_of_birth`) are not usually encapsulated into a single discrete token. Therefore, we might consider multiple `[MASK]` tokens for each relation label in the prompted input and define the verbalizer $\mathcal{M} : \mathcal{V} \rightarrow \mathcal{Y}$, such that it maps a set of predicted words in $\mathcal{V}$ for `[MASK]` tokens to actual relation labels $\mathcal{Y}$.

For instance, to apply a manually crafted verbalizer suggested by [13] to a given sentence $\mathbf{X}$ = {`Marcus Berg was born in Sweden`}, and the label $\mathbf{Y}$ = `per:country_of_birth`, we must assume multiple `[MASK]` tokens for this label. Thus, the input sentence $\mathbf{X}$ can be transformed into $\mathbf{X}_{prompt}$ = {`Marcus Berg was born in Sweden. [MASK] Marcus Berg [MASK] [MASK] [MASK] [MASK] Sweden.`}. Considering the predicted words as {`Marcus Berg was born in Sweden. person Marcus Berg was born in country Sweden`}, the verbalizer $\mathcal{M}$ should map the predicted words $v_i \in \mathcal{V}$ = `[person, was, born, in, country]` to the relation label `per:place_of_birth`.

This challenge encourages the exploration of trainable and adaptable verbalizers as alternative methods to overcome the above limitations and align more effectively in RE tasks [12, 21]. A solution proposed by KnowPrompt [6] suggests that instead of mapping multiple masked tokens to one actual relation label, consider *virtual label words* as special tokens and make a one-to-one mapping between the virtual label words and the actual relation labels. The virtual label words are tokens not defined in the vocabulary. These are trainable tokens that we define and integrate into the vocabulary so SLM can learn to represent them. We consider these virtual tokens $\mathcal{V}_c = \{v_1, \cdots, v_m\}$ as a subset of $\mathcal{V}$, where m represents the number of relation labels. Each $v_i \in \mathcal{V}_c$ is a virtual label word within the continuous vocabulary space. The optimization of these virtual label words involves the adjustment of the weights within the word-embedding layer of the SLM $\mathcal{S}$. We initialize $\mathcal{V}_c$ by averaging the tokens in each relation label. This initial setup may provide a more knowledgeable starting point for the verbalization process [6].

3.2.3 ***Training Objective.*** The fine-tuning process involves two optimization stages: (1) optimizing the virtual label words and (2) optimizing the SLM $\mathcal{S}$ parameters. During the first stage, we optimize the virtual label words by maximizing probability distribution $p(\mathcal{V}_c(\texttt{[MASK]}) \rightarrow \mathbf{Y}' | \mathbf{X}_{prompt}; \mathbf{Y}' \in \mathcal{Y})$, where $\mathcal{V}_c(\texttt{[MASK]})$ is the masked virtual word, $\mathbf{Y}'$ is the predicted label, and $\mathbf{X}_{prompt}$ indicates the prompted input sentence. We optimize this by minimizing the cross-entropy loss between the ground-truth label $\mathbf{Y}$ and the predicted label $\mathbf{Y}'$. Subsequently, after acquiring optimal virtual

[3] https://en.wikipedia.org/

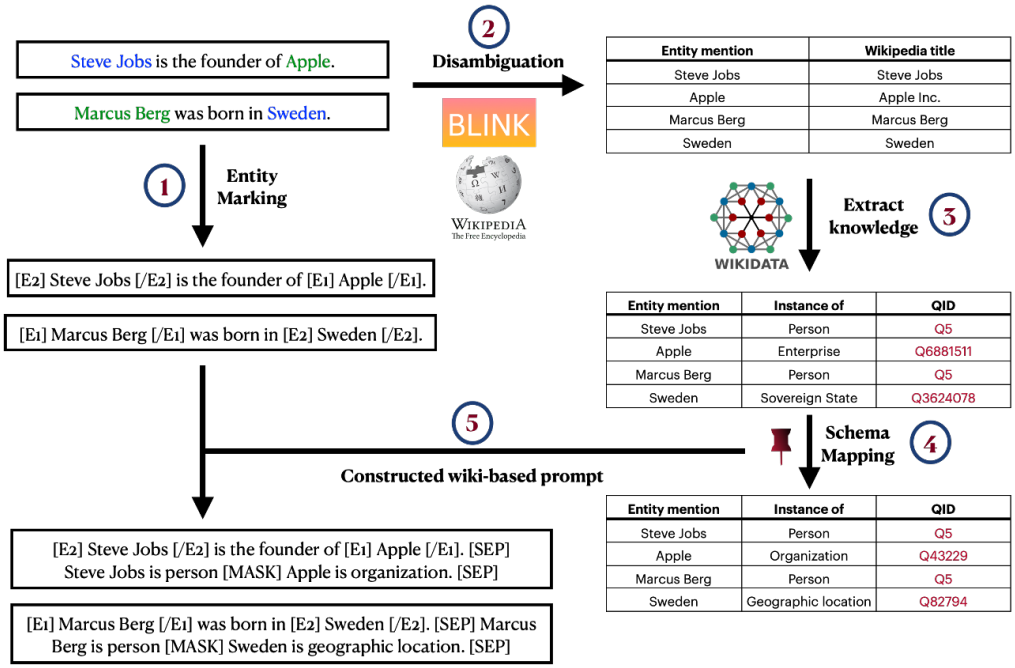

Figure 2: Overview of wiki-based prompt construction for prompt-tuning of SLMs; QID refers to the unique ID of items within Wikidata.

```
Instruction:
Use the context below with a pair of subject and object entities to
decide the relationship between subject and object entities
from candidate relations: [Enumerated list of relations]
Context:
[E1]Marcus Berg[/E1] [type]person[type] was born in [E2]Sweden[/E2]
[type]geographical location[type].
Response:
[Marcus Berg, born in country, Sweden]
```

```
Instruction:
Use the context below with a pair of subject and object entities to
decide the relationship between subject and object entities
from candidate relations: [Enumerated list of relations]
Context:
[E2]Martin Lorentzon[/E2] [type]person[type] is the co-founder of
[E1]Spotify[/E1] [type]organization[type].
Response:
[Spotify, founded by, Martin Lorentzon]
```

Figure 3: Illustration of instruction data for RE. The text highlighted in red represents the list of pre-defined relation labels in the natural language format. Tokens highlighted in blue color indicate the markers showing the subject and object entities and their corresponding types.

label words from the preceding optimization stage, we utilize the same loss function to fine-tune all the $\mathcal{S}$ parameters.

3.3 Instruction-tuned LLMs Using Wiki-based Prompts

Due to the challenges of prompt-tuning SLMs for RE, including verbalizer creation and optimization, we extended our exploration to instruction-tuned LLMs to advance RE tasks with wiki-based prompts. This section summarizes the methodology for incorporating instruction-following LLMs into our solution for the RE task. We discuss the integration of wiki-based prompts (discussed in Section 3.1) and detail the creation of an instruction RE dataset from the standard RE dataset. We then describe the subsequent SFT of Llama 2 [39] using PEFT. Finally, we investigate DPO and aligning SFT of LLMs using the RL-based SFT step with human preference data.

3.3.1 ***Creating Instruction Data.*** A crucial step in instruction-tuning is creating an instruction RE dataset to fine-tune LLMs on RE downstream tasks [31]. As discussed in Section 2.2.1, aligned with similar works on RE instruction tuning [44], we craft the instruction data by considering three items in each data example: (1) *Instruction* that describes the RE task, (2) *Context*, which is the entity marked input sentence with information of subject and object entities sourced from Wikidata (see Secteion 3.1), and (3) *Response*, which is the desired response indicating the subject and object entities and the relation label between them. We create the *Response* part of the data by transforming the original relation labels into their natural language equivalents. By providing ChatGPT [4] with a list of original relation labels, such as `[org:founded]` and `[per:employee_of]`, we generate coequals such as `[founded in]` and `[work for]`. Figure 3 depicts two examples of instruction data where the *instance_of* attributes associated with subject and object entities are incorporated into the context. In the instruction data the subject and object entities are represented by `[E1]` and `[E2]` tokens, respectively. Additionally, the *instance_of* attributes associated with these entities are highlighted using a distinct `[type]` token.

3.3.2 ***Instruction SFT of LLMs.*** Instruction-tuned LLMs are expansive language models that undergo SFT to tailor their responses to specific instructions [31]. For SFT of LLMs, we should first provide it with annotated instruction RE data, constructed in Section 3.3.1. However, fine-tuning all parameters of an LLM is computationally expensive; thus, we need an alternative solution to

[4] https://openai.com/blog/chatgpt

fine-tune only a subset of LLM parameters without sacrificing performance. To this end, we applied LoRA [16] to SFT Llama 2 [39] on instruction RE data.

LoRA decomposes the LLM's weight update matrix into low-dimensional matrices without losing crucial information. For example, let ΔW represent the weight update for an $A \times B$ weight matrix. This update can be decomposed into two matrices: $\Delta W = W_A W_B$, where W_A is a $A \times r$-dimensional matrix and W_B is a $r \times B$-dimensional matrix. Here, r signifies the rank (reduced vector) dimension, which is considerably smaller than the dimension of the model's parameters. LoRA SFT adopts a low-rank strategy, where in the low-rank context, matrices contain redundant rows or columns. Therefore, instead of updating all the model weights, LoRA SFT maintains the LLM's parameters W untouched and solely focuses on training the rank-decomposition matrices A and B. This approach effectively reduces memory consumption and facilitates the efficient fine-tuning of LLMs.

The specific steps of LoRA SFT are as follows: First, the LLM parameters are projected onto a lower-dimensional subspace using principal component analysis (PCA). Then, the projected parameters are fine-tuned using the instruction RE dataset. Finally, the fine-tuned parameters are used to predict the relations in the test dataset. A supervised learning algorithm, such as stochastic gradient descent, optimizes projected parameters. The loss function for an SFT algorithm is typically the cross-entropy loss between the predicted and ground-truth relations. Although LoRA SFT can improve the performance of LLMs on RE instruction tuning, further alignment of the SFT of LLM with human preference data through another SFT step is still necessary to achieve the most accurate results.

3.3.3 ***DPO Training.*** Although SFT assists LLMs in understanding the semantic meaning of prompts and generates meaningful responses, the SFT focuses solely on instructing the model about optimal responses and does not offer guidance on suboptimal alternatives [42]. Therefore, in addition to LoRA SFT, we also applied DPO [32] to align the LLM with human preference responses. To apply DPO, we first need to add the dispreferred responses to the dataset. To do so, we call the pair of text responses $\mathbf{Y}_j$ and $\mathbf{Y}_K$ as human preference data since one response is preferred to the other by a human evaluator ($\mathbf{Y}_j \gg \mathbf{Y}_k$). Regarding relation extraction, we assume the preferred responses $\mathbf{Y}_j$ as ground-truth relation labels. To achieve responses humans do not prefer, we assigned a wrong and noisy response as dispreferred one $\mathbf{Y}_k$ to each training example where the wrong response refers to the responses with wrong relation labels.

DPO does not require constructing an explicit reward function. Instead, it measures how well the model aligns with the preference dataset created by SFT LLM, as the reference model trained on the ground-truth data (the model trained with LoRA SFT in Section 3.3.2). In other words, instead of training a reward function, DPO directly optimizes a pre-trained LLM to maximize the likelihood of generating responses that humans prefer by using the SFT model as a reference model. The DPO reward is the difference between pre-trained and SFT of LLM's generated response. This allows us to skip the reward modeling step and directly use the preference model (SFT LLM) to optimize the pre-trained LLM. It is worth mentioning that the gradient of the loss function increases the likelihood of the preferred response $\mathbf{Y}_j$ and decreases the likelihood of the dispreferred response $\mathbf{Y}_k$.

4 EVALUATION

In this section, we conduct experiments to measure and compare the effectiveness of wiki-based prompts in fine-tuning both SLMs and LLMs on downstream RE tasks across different scenarios, including *standard RE* and *few-shot RE*.

4.1 Datasets and Implementation Details

We conducted our experiments using four distinct English-language RE datasets, which are TACRED [49], TACREV [1], RE-TACRED [37], and SemEval-2010 Task8 (SemEval) [14]. TACRED is a widely recognized RE dataset comprising 42 relation labels, including a label for cases where no specific relation exists between subject and object entities. TACREV is derived from TACRED and includes re-labeled validation and test datasets while retaining the same training data. RE-TACRED is a modified version of TACRED, which is re-annotated with 40 labels. Finally, SemEval specializes in classifying semantic relations between pairs of nominals, such as `apple` and `fruit`, encompassing 19 possible relations.

We performed our experiments using an NVIDIA A100-SMX4-40GB GPU. For SLM, we employed RoBERTa-large [26], a pre-trained LM with 123 million parameters, and for LLM, we utilized Llama 2-7b [39], a model with 7 billion parameters. All these models are available at the Hugging Face page[5]. To apply LoRA SFT on Llama 2, we applied the following hyper-parameter setting: $\alpha = 16$ as the scaling factor for the low-rank matrices, a dropout rate of 0.1 for the dropout probability of the LoRA layers, and a dimension $r = 64$ for the low-rank matrices. The learning rate was $2e-4$, and the number of training epochs was 3. Moreover, the maximum sequence length was 2048, and all optimizations were performed using the AdamW optimizer with a warm-up ratio of 3%.

4.2 Baselines

In our experiments, we compared our approach against several RE frameworks to evaluate our approach's effectiveness in RE tasks. As baselines, we consider the following RE frameworks:

- Standard fine-tuning: (1) SpanBERT [18], a span-based pre-trained LM designed to represent and predict text spans and is fine-tuned on RE downstream tasks, (2) LUKE [45], a pre-trained LM incorporates an entity-aware self-attention layer to generate contextually rich word representations. This pre-trained model is fine-tuned using the RE downstream task, and (3) TYP Marker [51], an RE framework that enhances performance using entity-typed markers during the fine-tuning SLMs.
- Prompt tuning: (1) KnowPrompt [6], a prompt-based RE framework that directly incorporates knowledge from relation labels into the prompt structure, enabling improved RE performance, and (2) PTR [13], a prompt-based RE framework that applies logic rules to construct prompt templates with different sub-prompts.

[5] https://huggingface.co/models

Table 1: RE methods and acronyms.

Acronym	Methodology
SpanBERT	Fine-tuned SpanBERT language model for RE tasks.
LUKE	Fine-tuned LUKE language model for RE tasks.
TYP Marker	Fine-tuned RoBERTa on RE tasks with typed markers.
PTR	Rule-based Prompt-Tuning for RE.
KnowPrompt	Framework optimizing knowledge shared among relation labels in Prompt-Tuning for RE.
Wiki-Tuning RoBERTa	Prompt-Tuning approach for RE using wiki-based prompt construction.
Prompt-Tuning RoBERTa	Prompt-Tuning approach for RE without specific techniques for prompt construction.
Wiki-based KnowPrompt	Framework combining wiki-based prompts with knowledge optimization among relation labels for RE.
Wiki-SFT Instruction	Supervised instruction-tuning of Llama 2-7b with wiki-based prompt construction for RE.
Wiki-SFT DPO Instruction	Supervised instruction-tuning of Llama 2-7b with wiki-based prompt construction, followed by DPO fine-tuning for RE.
Instruction-tuning	Supervised instruction-tuning of Llama 2-7b for RE.
ICL-RE	In-context learning framework for RE.

- In-context learning: ICL-RE [44], a framework that leverages in-context learning and data generation techniques for few-shot RE using GPT-3.5.

4.3 Evaluation Metrics

As the evaluation metric, we employed *Micro-F1* used by the previous methods. However, due to the nature of instruction-tuned LLMs, which generate text spans, we specifically used the *span-based Micro-F1* [10], where a predicted relation is considered correct if the generated relation label matches the ground-truth relation label and the model accurately predicts the text spans corresponding to the subject and object entities. This evaluation approach ensures that the relation label and the precise boundaries of the subject and object entities are considered when assessing the model performance in RE.

4.4 Results and Comparison

This section presents the results of various models across different RE tasks, considering two distinct settings: *standard RE* and *few-shot RE*. Standard RE entails scenarios where a *rich-resource* RE dataset containing many annotated examples is available for model training, while few-shot RE involves training models using a *low-resource* RE dataset, where the availability of annotated examples in the training dataset is limited. To evaluate the performance of the models on the few-shot setting, we conduct random sampling of k instances (k-shot) for each relation label from each dataset, with k values set at 8, 16, and 32. It is crucial to mention that each randomly sampled k-shot dataset yields distinct results; thus, we present the average performance across five different randomly sampled datasets.

To enhance the clarity, we categorize the methods into four learning approaches:

- *Standard SFT*: Traditional fine-tuning of pre-trained SLMs on RE tasks.
- *Prompt-Tuning*: Fine-tuning pre-trained SLMs using prompts with `[MASK]` token for RE tasks.
- *Instruction-tuning*: Fine-tuning LLMs on instruction-based RE data to align model behavior.
- *In-context learning*: Utilizing prompts with few demonstrations as examples for few-shot RE.

Table 1 shows the model details and their acronyms. We use these acronyms throughout the results and comparison sections.

4.4.1 ***Standard RE.*** We initially assess the performance of fine-tuned SLMs, prompt-tuned SLMs, and instruction-tuned LLMs using wiki-based prompts on standard RE datasets. In Table 2, we present a comparative analysis of the results obtained from these wiki-based approaches and baseline models. Furthermore, we incorporate wiki-based prompts into the KnowPrompt framework [6] (*Wiki-based KnowPrompt*) to evaluate the efficacy of these informative prompts when applied to existing state-of-the-art models, which was only feasible for KnowPrompt among baseline models. It is important to note that fine-tuning the model in the ICL-RE [44] was not performed; thus, we cannot evaluate this framework within a standard RE setting.

As illustrated in Table 2, combining wiki-based prompts with KnowPrompt demonstrates superior performance compared to the other models, emphasizing the effectiveness of employing wiki-based prompts. Moreover, using wiki-based prompts enhances the performance of prompt-tuned SLMs and instruction-tuned LLMs compared to scenarios where they are not used. Furthermore, while wiki-based prompt-tuning on RoBERTa yields encouraging results on various RE datasets, its performance falls slightly short of the best-reported performances in some instances.

4.4.2 ***Few-shot RE.*** In the Few-shot RE evaluation, we conducted extensive assessments to measure the usefulness of different prompt-tuning approaches, including our wiki-based prompts. Since the wiki-based prompt is primarily a method for constructing prompts, it is applied within the prompt-tuning paradigm. Furthermore, we extend our evaluation to include instruction-tuning with wiki-based prompts and compare them with in-context learning and standard fine-tuning methods within the few-shot setting.

In Table 3, we present a comparative evaluation of the state-of-the-art frameworks, encompassing prompt tuning, standard fine-tuning, and instruction-tuning approaches. Here, we demonstrate our innovative wiki-based prompt construction in the context of few-shot RE. The results highlight the superiority of prompt-tuning methods over standard fine-tuning and instruction-tuning techniques. Specifically, KnowPrompt [6] consistently outperforms the baselines. However, the standout performer is our *Wiki-based KnowPrompt* model, which leverages wiki-based prompt construction in combination with relation label knowledge constraints from the KnowPrompt method. This synergy outperforms all the baselines on three datasets by +0.9, +1.97, and +0.24 Micro-F1 on average. These enhancements over the best-reported results demonstrate the substantial advantages of incorporating wiki-based prompt construction in the prompt tuning paradigm. Furthermore, our *Wiki-tuning RoBERTa* indicates promising results in few-shot RE. Notably, it accomplishes this without the need for complex rule-based sub-prompt construction or computationally expensive prompt optimization, unlike some other prompt tuning approaches.

In the instruction-tuning paradigm, our *Wiki-SFT instruction* indicates solid performance, averaging a Micro-F1 score from 24.4

Table 2: Standard RE results. Text in red represents the achieved enhancement of using wiki-based prompts over the results of KnowPrompt [6].

Learning Paradigm	Model	TACRED	TACREV	RE-TACRED	SemEval
Standard Supervised Fine-tuning (Micro-F1)	*SpanBERT* [18]	70.8	78.0	85.3	89.8
	LUKE [45]	72.7	80.6	-	-
	TYP Marker [51]	74.6	83.2	91.1	-
Prompt-Tuning (Micro-F1)	*PTR* [13]	72.4	81.4	90.9	89.9
	KnowPrompt [6]	72.4	82.4	91.3	**90.2**
	Wiki-Tuning RoBERTa	70.6	80.4	86.2	82.7
	Prompt-Tuning RoBERTa	64.3	72.8	79.5	83.2
	Wiki-based KnowPrompt	**76.8** (+4.4)	**84.1** (+1.7)	**91.8** (+0.5)	88.7
Instruction Tuning LLM (Span-based Micro-F1)	*Wiki-SFT Instruction*	57.9	60.3	63.1	66.3
	Wiki-SFT DPO Instruction	59.3	61.7	65.0	69.4
	Instruction-tuning	51.1	53.6	55.2	56.7

Table 3: Few-shot RE results. *AVG* indicates the averaged performance over the three few-shot settings and *AVG* indicates the averaged performance over the three few-shot settings. Text in red represents the achieved enhancement of using wiki-based prompts over the best-reported result.

Model		TACRED				TACREV				RE-TACRED				SemEval			
		k=8	k=16	k=32	AVG	k=8	k=16	k=32	AVG	k=8	k=16	k=32	AVG	k=8	k=16	k=32	AVG
Standard Supervised Fine-tuning (Micro-F1)	*SpanBERT* [18]	8.4	17.5	19.8	15.23	5.2	5.7	6.3	5.73	14.6	28.7	31.6	24.96	38.7	59.6	75.2	57.86
	TYP Marker [51]	26.5	29.9	30.2	28.86	26.7	29.5	31.4	29.3	44.8	54.1	58.3	52.4	-	-	-	-
Prompt-Tuning (Micro-F1)	*PTR* [13]	28.1	30.7	32.1	30.30	28.7	31.4	32.4	30.83	51.5	56.2	62.1	56.6	70.5	81.3	84.2	78.66
	KnowPrompt [6]	30.7	31.9	33.7	32.1	31.7	33.1	34.7	33.16	55.3	**63.3**	65.0	**61.2**	74.3	82.9	**84.8**	80.66
	Wiki-Tuning RoBerta†	24.8	26.1	29.4	26.76	27.6	30.1	33.5	30.4	43.7	51.4	59.7	51.60	56.3	69.7	73.1	66.36
	Prompt-Tuning RoBERTa†	21.6	24.2	27.5	24.43	23.2	25.1	29.3	25.86	32.6	36.7	41.4	36.9	50.30	56.5	68.2	58.33
	Wiki-based KnowPrompt	**31.2**	**32.6**	**35.2**	**33.00** (+0.9)	**33.4**	**35.7**	**36.3**	**35.13** (+1.97)	**55.9**	61.4	**65.2**	60.83	**75.7**	**83.6**	83.4	**80.90** (+0.24)
Instruction Tuning (Span-based Micro-F1)	*Wiki-SFT Instruction*	17.6	26.5	29.1	24.4	19.2	30.9	32.6	27.56	22.1	38.7	42.3	34.36	30.2	39.8	44.3	38.1
	Instruction-tuning	16.7	19.3	23.7	19.90	17.0	26.6	29.1	24.23	17.7	26.3	32.5	25.50	26.4	31.2	37.9	31.83
	Wiki-SFT DPO Instruction	23.3	29.7	33.3	28.76	26.0	31.9	34.5	30.80	33.8	46.2	49.1	43.03	29.8	33.4	38.1	33.76
In-context Learning (Span-based Micro-F1)	*ICL-RE (5-shot)* [44]			27.8				-				34.0				39.4	

to 38.1 across different datasets. This underscores the value of the wiki-based prompt construction, particularly when compared to the instruction-tuning lacking wiki-based prompts. Meanwhile, it can be observed that DPO optimization improves instruction-tuning by around 4 to 9 percent on average, indicating the effectiveness of aligning LLMs with human-preferred responses. Overall, these results collectively emphasize the potential of incorporating external knowledge, particularly wiki-based prompts, to enhance few-shot RE models significantly.

4.4.3 ***Training Time.*** As illustrated in Figure 4, the training run time for different systems in our experiments provides valuable insights into the computational demands for each approach. Among these, the *Wiki-SFT instruction 8-shot* model, tuned on the TACRED 8 shots dataset, completes in the shortest time at 483.89 seconds. The *Wiki-SFT instruction 16-shot* and *Wiki-SFT instruction 32-shot* models closely follow, taking 953.18 and 1170.64 seconds, respectively. The longer durations are attributed to larger training datasets, with 16 and 32 examples per relation label.

The *Wiki-tuning RoBERTa* model emphasizes the integration of wiki-based prompts in the prompt-tuning paradigm, resulting in a longer training time of 13,575.44 seconds due to verbalizer optimization. The *Wiki-SFT DPO instruction* model, involving additional fine-tuning with human-preferred responses, exhibits an even more extended training period, lasting 16,016.31 seconds. This increased duration can be attributed to the direct preference optimization process, which refines the model through multiple iterations, aligning it with human-preferred responses. Finally, the *Wiki-based KnowPrompt* model has the longest training time, totaling 22,879.42 seconds, highlighting its computational intensity due to prompt template and verbalizer optimization. These extended training times emphasize the trade-offs between superior model performance and increased computational burden.

Figure 4: Training Time

4.5 Discussion and Limitations

The results demonstrate the effectiveness of incorporating wiki-based prompts in both prompt-tuning and instruction-tuning approaches for RE tasks. In standard RE evaluation, when combined with the KnowPrompt framework, wiki-based prompts outperform other models, highlighting their utility. In few-shot RE evaluation, prompt-tuning consistently outperforms standard fine-tuning and instruction-tuning, with the *Wiki-based KnowPrompt* model achieving remarkable results. However, it is worth noting that instruction-tuned LLMs, while not requiring the design of verbalizers as generative models, still face challenges in outperforming other approaches in classification tasks like RE.

A significant obstacle is *aleatoric uncertainty* caused by class definition overlap. This problem occurs when the semantics we use for labels are not well-defined, and the model has difficulty distinguishing between similar labels having similar semantics. For instance, labels "`born in city`" and "`born in country`". Although

DPO fine-tuning shows potential in enhancing SFT instruction-tuned LLMs, it does not outperform other models and incurs higher computational costs and longer training times.

Moreover, addressing another challenge associated with instruction-tuned LLMs related to the quality of instruction-based RE datasets is crucial. The performance of instruction-tuned LLMs relies heavily on the quality and specificity of the provided instructions. Even a single alteration in the instruction prompt can yield substantial differences in results. Additionally, it is crucial to acknowledge challenges encountered in entity disambiguation, which can result in incorrect categorization. Managing null values for the *instance_of* attribute in Wikidata is a significant challenge. This limitation has resulted in notable inconsistencies in entity categorization, significantly impacting performance in tasks such as SemEval, where a substantial portion of Wikidata knowledge comprises entity descriptions. This limitation highlights a broader concern regarding the reliance on external knowledge bases for NLP tasks. Addressing this issue underscores the necessity for ongoing research in refining entity disambiguation techniques to enhance model performance in tasks reliant on external knowledge sources.

5 RELATED WORK

In this section, we first review the existing related work in RE and then explore in more detail the literature that takes advantage of prompt-tuning and instruction-tuning for RE.

5.1 Relation Extraction

RE has been the subject of extensive research in NLP. A popular approach for RE has been the rule-based systems that use manually crafted patterns and heuristics [30]. Earlier works explored neural architectures such as BiLSTM [38], and RNN [7], indicating potential in capturing relationships. However, they often require substantial labeled data, which can be scarce in specific domains. Recently, pre-trained LMs (PLMs) have shown significant improvements in RE by applying transformer-based architectures as the backbone for learning text representation [36]. Another RE framework [51] fine-tunes the transformer-based models with entity-typed markers to achieve better results on RE tasks. Despite the satisfactory performance of PLMs in RE tasks, these approaches have limited generalization capability in few-shot RE tasks.

5.2 Prompt-tuning

Recently, the concept of prompt-tuning, initiated with GPT-3 [5], emerged to connect pre-training and fine-tuning objectives [11, 24, 25]. These methods reframe downstream tasks using textual templates that align input sentences with pre-training examples, enabling better knowledge transfer. According to [19], a well-chosen prompt can be as effective as hundreds of data points, making prompt-tuning advantageous for few-shot tasks. Optimal performance in this learning paradigm requires precise prompt designing and selecting a set of label words (a.k.a verbalizer) [34].

PTR [13] is a prompt-tuning framework for RE tasks. PTR uses logic rules to construct prompts automatically by combining multiple sub-prompts and incorporating a manually crafted verbalizer. Despite the success of PTR on few-shot RE, creating logic rules is domain-dependent, demanding domain expertise and knowledge to formulate these rules tailored to each domain. On the other hand, various studies suggested that incorporating knowledge about subject and object entities in RE tasks can substantially enhance the model's performance [2, 20, 46]. Consequently, KnowPrompt [6], another RE model, explored integrating knowledge inside the relation labels into prompt creation. This approach creates virtual entity-type tokens, specifying subject and object scopes based on token frequencies in relation labels in the training dataset. These tokens are optimized during two training stages for knowledge-infused prompts.

Another approach is proposed by Liu et al. [23] to generate knowledge prompts by employing a language model to extract knowledge from the input text, subsequently using this acquired knowledge to formulate the prompt. In contrast to these models that rely on language models for the computationally expensive and potentially unreliable task of generating or refining knowledge, we leverage the extensive knowledge stored within the knowledge base (e.g., Wikipedia or Wikidata) to construct informative prompts for RE tasks automatically.

Furthermore, Brate et al. [33] explore the effects of enriching prompts with additional contextual information leveraged from the Wikidata knowledge graph on language model performance. They specifically compare the performance of naive vs. knowledge graph-engineered cloze prompts for entity genre classification in the movie domain and enrichment of cloze-style prompts. In our study, we extend this exploration by expanding the use of wiki-based prompts to both prompt-tuning and instruction-tuning RE models, demonstrating the effectiveness of this approach across different RE settings.

5.3 Instruction-tuning

In recent years, LLMs like GPT-3 [5], and Llama 2 [39] have shown remarkable progress across various NLP tasks. One approach to aligning LLMs to the user's expectation is instruction-tuning, where the LLM is fine-tuned on pairs of human instructions and desired outputs [31, 50]. Within the realm of RE, there is currently no research that directly employs instruction-tuning for RE tasks. However, some studies have utilized instruction-tuning for Information Extraction [8, 10, 40]. For instance, UIE [41] transforms IE tasks into a seq2seq format and addresses them by fine-tuning the 11B FlanT5 model [8] on the constructed instruction-based dataset. Nevertheless, it is essential to note that a direct comparison between this framework and our work is not feasible. This is primarily due to the pre-annotation of entity types in sentences in the aforementioned framework. Our study employs our wiki-based approach to construct informative prompts, specifically focusing on its effectiveness in constructing high-quality instruction-based RE data.

6 CONCLUSION

This paper has introduced a novel approach of wiki-based prompts as a prompt construction approach to enhance Relation Extraction (RE) tasks by leveraging external knowledge from Wikidata to craft informative prompts, called *wiki-based* prompt, for prompt-tuning and instruction-tuning of language models. Our findings demonstrate the effectiveness of incorporating wiki-based prompts in both prompt-tuning and instruction-tuning approaches, with the *Wiki-based KnowPrompt* model standing out as a considerable achievement in the few-shot RE evaluation. However, our study also

revealed essential challenges and limitations in this field, including aleatoric uncertainty due to relation label definition overlap, the quality of instruction-based RE dataset, and accurate entity disambiguation. In conclusion, our work represents a significant step forward in improving RE tasks using external knowledge sources. Nevertheless, it also underscores the need for ongoing research and refinement in addressing the aforementioned challenges and limitations. Future research in this area should focus on reducing prediction uncertainty, enhancing the quality of instruction-based datasets, and refining entity disambiguation techniques to utilize the full potential of wiki-based prompts in advancing the capabilities of language models in RE tasks and beyond. Additionally, further investigations should be conducted to evaluate the impact of incorporating other knowledge bases such as DBpedia [6] into the prompt construction to refine language models.

REFERENCES

[1] Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task. In *Proceedings of the 58th Annual Meeting of the ACL Conference*.

[2] Anson Bastos and Nadgeri. 2021. RECON: relation extraction using knowledge graph context in a graph neural network. In *Proceedings of the Web Conference 2021*. 1673–1685.

[3] Matthias Baumgartner, Wen Zhang, and Paudel. 2018. Aligning knowledge base and document embedding models using regularized multi-task learning. In *The Semantic Web–ISWC 2018*. Springer, 21–37.

[4] Emily M Bender, Timnit Gebru, and Angelina McMillan-Major. 2021. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM FAccT conference*. 610–623.

[5] Tom Brown, Benjamin Mann, Nick Ryder, et al. 2020. Language models are few-shot learners. *NeurIPS* 33 (2020), 1877–1901.

[6] Xiang Chen, Ningyu Zhang, et al. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[8] Hyung Won Chung et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022).

[9] Jacob Devlin, Ming-Wei Chang, et al. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] Yaojie Lu et al. 2022. Unified structure generation for universal information extraction. (2022). https://doi.org/10.18653/v1/2022.acl-long.395

[11] Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.295

[12] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. (2021). https://doi.org/10.18653/v1/2021.acl-long.381

[13] Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2022. Ptr: Prompt tuning with rules for text classification. *AI Open* 3 (2022), 182–192.

[14] Iris Hendrickx et al. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. 33–38.

[15] hirui Pan et al. 2023. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *arXiv preprint arXiv:2306.08302* (2023).

[16] Edward J Hu, yelong shen, et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR 2022*. https://openreview.net/forum?id=nZeVKeeFYf9

[17] Shengding Hu et al. 2022. Knowledgeable Prompt-tuning: Incorporating Knowledge into Prompt Verbalizer for Text Classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 2225–2240.

[18] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics* (2020).

[19] Teven Le Scao and Alexander M Rush. 2021. How many data points is a prompt worth?. In *Proceedings of the 2021 Conference of NAACL*. 2627–2636.

[20] Pengfei Li, Kezhi Mao, Xuefeng Yang, and Qi Li. 2019. Improving relation extraction with knowledge-attention. *arXiv preprint arXiv:1910.02724* (2019).

[21] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190* (2021).

[22] Belinda Z Li et al. 2020. Efficient one-pass end-to-end entity linking for questions. *arXiv preprint arXiv:2010.02413* (2020).

[23] Jiacheng Liu, Alisa Liu, et al. 2022. Generated Knowledge Prompting for Commonsense Reasoning. In *Proceedings of the 60th Annual Meeting of the ACL*.

[24] Pengfei Liu et al. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* (2023).

[25] Xiao Liu, Kaixuan Ji, Yicheng Fu, et al. 2022. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-short.8

[26] Yinhan Liu, Myle Ott, Naman Goyal, et al. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[27] Denis Lukovnikov, Asja Fischer, and Jens Lehmann. 2019. Pretrained transformers for simple question answering over knowledge graphs. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference*. Springer.

[28] Da Luo, Jindian Su, and Shanshan Yu. 2020. A BERT-based approach with relation-aware attention for knowledge base question answering. In *2020 IJCNN*. IEEE.

[29] Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv preprint arXiv:2104.08773* (2021).

[30] R Mooney. 1999. Relational learning of pattern-match rules for information extraction. In *Proceedings of the 16th national conference on artificial intelligence*.

[31] Long Ouyang, Jeffrey Wu, Xu Jiang, et al. 2022. Training language models to follow instructions with human feedback. *NeurIPS* 35 (2022), 27730–27744.

[32] Rafael Rafailov, Archit Sharma, Eric Mitchell, et al. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290* (2023).

[33] Brate Ryan, et al. 2022. Improving Language Model Predictions via Prompts Enriched with Knowledge Graphs. In *DL4KG ISWC2022*.

[34] Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically Identifying Words That Can Serve as Labels for Few-Shot Text Classification. International Committee on Computational Linguistics. https://aclanthology.org/2020.coling-main.488

[35] Timo Schick and Hinrich Schütze. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.20

[36] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158* (2019).

[37] George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 13843–13850.

[38] Julien Tourille, Olivier Ferret, Aurelie Neveol, et al. 2017. Neural architecture for temporal relation extraction: A Bi-LSTM approach for detecting narrative containers. In *Proceedings of the 55th Annual Meeting of the ACL Conference*.

[39] Hugo Touvron et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[40] Zhen Wan, Fei Cheng, et al. 2023. GPT-RE: In-context Learning for Relation Extraction using Large Language Models. In *The 2023 EMNLP Conference*. https://openreview.net/forum?id=mTiHLHu3sP

[41] Xiao Wang, Weikang Zhou, et al. 2023. InstructUIE: Multi-task Instruction Tuning for Unified Information Extraction. *arXiv preprint arXiv:2304.08085* (2023).

[42] Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966* (2023).

[43] Zeqiu Wu, Yushi Hu, et al. 2023. Fine-Grained Human Feedback Gives Better Rewards for Language Model Training. *arXiv preprint arXiv:2306.01693* (2023).

[44] Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023. How to Unleash the Power of Large Language Models for Few-shot Relation Extraction? Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.sustainlp-1.13

[45] Ikuya Yamada and Asai Akari et al. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.523

[46] Hongbin Ye, Ningyu Zhang, Shumin Deng, Xiang Chen, Hui Chen, Feiyu Xiong, Xi Chen, and Huajun Chen. 2022. Ontology-enhanced Prompt-tuning for Few-shot Learning. In *Proceedings of the ACM Web Conference 2022*. 778–787.

[47] Ningyu Zhang, Xiang Chen, Xin Xie, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, Luo Si, and Huajun Chen. 2021. Document-level relation extraction as semantic segmentation. *arXiv preprint arXiv:2106.03618* (2021).

[48] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. 2023. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199* (2023).

[49] Yuhao Zhang, Victor Zhong, et al. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of EMNLP Conference*.

[50] Shengyu Zhang et al. 2023. Instruction Tuning for Large Language Models: A Survey. *arXiv preprint arXiv:2308.10792* (2023).

[51] Wenxuan Zhou and Muhao Chen. 2022. An Improved Baseline for Sentence-level Relation Extraction. In *Proceedings of the ACL*. 161–168.

[6] https://www.dbpedia.org/