

RE-Matching: A Fine-Grained Semantic Matching Method for Zero-Shot Relation Extraction

Jun Zhao^{1*}, Wenyu Zhan^{1*}, Xin Zhao¹, Qi Zhang^{1†}, Tao Gui^{2†},
Zhongyu Wei³, Junzhe Wang¹, Minlong Peng⁴, Mingming Sun⁴

¹School of Computer Science, Fudan University

²Institute of Modern Languages and Linguistics, Fudan University

³School of Data Science, Fudan University

⁴Cognitive Computing Lab Baidu Research

{zhaoj19, qz, tgui}@fudan.edu.cn, wyzhan21@m.fudan.edu.cn

Abstract

Semantic matching is a mainstream paradigm of zero-shot relation extraction, which matches a given input with a corresponding label description. The entities in the input should exactly match their hypernyms in the description, while the irrelevant contexts should be ignored when matching. However, general matching methods lack explicit modeling of the above matching pattern. In this work, we propose a fine-grained semantic matching method tailored for zero-shot relation extraction. Following the above matching pattern, we decompose the sentence-level similarity score into entity and context matching scores. Due to the lack of explicit annotations of the redundant components, we design a feature distillation module to adaptively identify the relation-irrelevant features and reduce their negative impact on context matching. Experimental results show that our method achieves higher matching F_1 score and has an inference speed 10 times faster, when compared with the state-of-the-art methods.

1 Introduction

Relation extraction (RE) is a fundamental task of natural language processing (NLP), which aims to extract the relations between entities in unstructured text. Benefitting from high-quality labeled data, neural relation extraction has achieved superior performance (Han et al., 2020; Wu and He, 2019a; Zhao et al., 2021). However, it is expensive and even impractical to endlessly label data for a fast-growing number of new relations.

In order to deal with the emerging new relations that lack labeled data, zero-shot relation extraction (ZeroRE) has recently attracted more attention. Levy et al. (2017) frame the ZeroRE as a slot-filling task solved in a question-answering way. Each relation is associated with a few question templates.

*Equal Contributions.

†Corresponding authors.

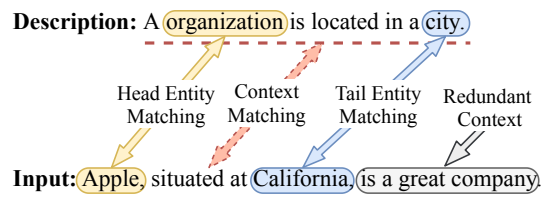


Figure 1: An example of the matching pattern of relational data. The input sentence contains a given entity pair, which should match the corresponding hypernyms (usually the entity type). The context describes the relations between entities, containing relation-irrelevant redundant information, which should be ignored when matching.

However, the templates are expensive and time-consuming to build (Chen and Li, 2021). Obamuyide and Vlachos (2018) simplify the templates to readily available relational descriptions and reformulate ZeroRE as a semantic matching task. Recently, pretrained model based ZeroRE methods have achieved great success. *Siamese scheme* and *full encoding scheme* are two mainstream methods for matching semantics. The siamese scheme separately encodes the input and description (Chen and Li, 2021). Therefore, the encoded representations of descriptions can be both stored and reused for each input, resulting in a fast inference. However, insufficient interaction during encoding also limits the matching performance. By contrast, the full encoding scheme performs self-attention over the pair to enrich interaction (Sainz et al., 2021), although the performance increase comes with a computational overhead. (For m inputs and n descriptions, the siamese scheme requires $m + n$ encodings, while the number is $m \times n$ for full encoding). An approach that combines the advantages of both can be attractive.

Unlike ordinary sentence pairs, relational data has a unique matching pattern, which is not explicitly considered by general matching methods. As shown in fig. 1, the entities in the input should

exactly match their hypernyms in the description (e.g., Apple and organization). Meanwhile, not all contextual words contribute equally to the relation semantics. For example, the clause “is a great company” is only used to modify Apple instead of expressing the relationship between Apple and California. Such redundant components should be ignored when matching. Due to the lack of explicit annotations to the redundant components, it is non-trivial for the model to learn to identify them.

In this work, we propose a fine-grained semantic matching method that improves both the accuracy and speed over the current state-of-the-art. Specifically, we decouple encoding and matching into two modules. While the encoding module follows a siamese scheme for efficiency, the matching module is responsible for the fine-grained interaction. Following the matching pattern of relational data, the sentence-level similarity score is decomposed into two: entity matching and context matching scores. To deal with the redundant components without explicit annotations, we design a feature distillation module. Context features that maximize a classification loss are identified as relation-irrelevant features. Then, the context representations are projected into the orthogonal space of the features to improve context matching. Experimental results show that this method outperforms state-of-the-art (SOTA) methods for ZeroRE, in terms of both accuracy and speed. Our codes are publicly available*.

The main contributions are three-fold: (1) We propose a fine-grained semantic matching method for ZeroRE, which explicitly models the matching pattern of relational data; (2) We propose a context distillation method, which can reduce the negative impact of irrelevant components on context matching; (3) Experimental results show that our method achieves SOTA matching F_1 score together with an inference speed 10 times faster.

2 Related Works

Text semantic matching aims to predict a matching score that reflects the semantic similarity between a given pair of text sequences. The approaches used to calculate the matching score roughly fall into one of two groups. The first group uses a siamese scheme, which separately maps the pair of text sequences into a common feature space wherein a dot product, cosine, or parameterized

non-linearity is used to measure the similarity (Bai et al., 2009; Huang et al., 2013; Zamani et al., 2018; Wu et al., 2018; Chen and Li, 2021). The second group adopts a full encoding scheme, in which the concatenation of the text pair serves as a new input to a nonlinear matching function. Neural networks with different inductive biases are used to instantiate the matching function (Wu et al., 2017; Yang et al., 2018; Zhang et al., 2018). More recently, large-scale pretrained language models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) are introduced to yield richer interactions between the text pair, to improve semantic matching.

Unlike the general matching methods, the proposed method is designed to be specifically used for relational data. By explicitly modeling the matching pattern, our method achieves SOTA performance while decreasing the computational cost by an order of magnitude compared with a full encoding scheme (as the more candidate relations exist, the more cost decreases).

3 Approach

In this work, we propose a fine-grained semantic matching method tailored for relational data. To facilitate inference efficiency, we adopt a siamese scheme to separately encode the input and the candidate relation descriptions. To explicitly model the matching pattern of relational data, we decompose the matching score into entities and contexts matching scores. In addition, we design a context distillation module to reduce the negative impact of irrelevant components in the input on context matching.

3.1 Task Formulation and Method Overview

Task Formulation: In Zero-shot relation extraction (ZeroRE), the goal is to learn from the seen relations $\mathcal{R}_s = \{r_1^s, r_2^s, \dots, r_n^s\}$ and generalize to the unseen relations $\mathcal{R}_u = \{r_1^u, r_2^u, \dots, r_m^u\}$. Such two sets are disjoint, i.e., $\mathcal{R}_s \cap \mathcal{R}_u = \emptyset$ and only the samples of the seen relations \mathcal{R}_s are available at the training phase. Following Chen and Li (2021), we formulate ZeroRE as a semantic matching problem. Specifically, given the training set $\mathcal{D} = \{(x_i, e_{i1}, e_{i2}, y_i, d_i) | i = 1, \dots, N\}$ with N samples, consisting of input instance x_i , target entity pair e_{i1} and e_{i2} , relation $y_i \in \mathcal{R}_s$ and its corresponding relation description d_i , we optimize a matching model $\mathcal{M}(x, e_1, e_2, d) \rightarrow s \in \mathbb{R}$, where the score s

*<https://github.com/zweny/RE-Matching>

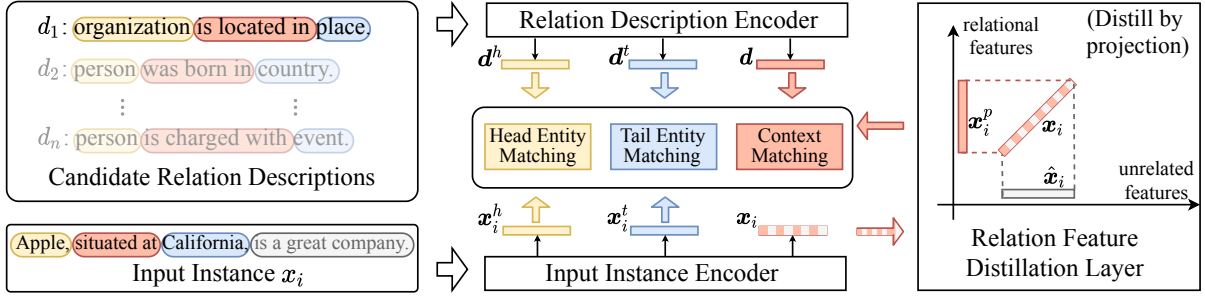


Figure 2: Overview of the proposed RE-Matching method. The input instance and candidate relation descriptions (left side) are separately encoded for efficiency. To model the matching pattern of relational data, we calculate similarity by entity and context matching (middle). In addition, we design a distillation module to reduce the impact of irrelevant components (the gray part in the input) on context matching (right side).

measures the semantic similarity between the input instance x and the relation description d . When testing, we transfer the matching model \mathcal{M} to extract unseen relations in \mathcal{R}_u . Specifically, given a sample (x_j, e_{j1}, e_{j2}) expressing an unseen relation in \mathcal{R}_u , we make prediction by finding the relation \hat{y}_j whose description has the highest similarity score with the input sample.

Method Overview: We approach the problem with a fine-grained semantic matching method tailored for relational data. As illustrated in fig. 2, we decouple encoding and matching into two modules, explicitly modeling the matching pattern of relational data while ensuring efficient inference.

The encoding module is designed to extract both entity and contextual relational information from an input instance x_i and a candidate description $d \in \{d_{r_i^s} | i = 1, \dots, n\}$, which lays the groundwork for fine-grained matching. To enrich the insufficient entity information in d , we separate the hypernyms (i.e. entity types) in d and design several methods to automatically expand them into a complete head (tail) entity description d^h (d^t). We adopt a fixed Sentence-BERT (Reimers and Gurevych, 2019) to encode the d , d^h , d^t to their representations \mathbf{d} , \mathbf{d}^h , $\mathbf{d}^t \in \mathcal{R}^n$, respectively. For input x_i , we use a trainable BERT (Devlin et al., 2019) to encode its context and entity representation x_i , x_i^h , $x_i^t \in \mathcal{R}^n$. Because descriptions and instances are separately encoded, the computational complexity is reduced from $\mathcal{O}(mn)$ to $\mathcal{O}(m+n)$, compared with the full encoding scheme (m, n represent the number of candidate descriptions and input instances, respectively).

The matching module is responsible for the interaction between input x_i and description d . The score of entity matching is directly calculated

with cosine similarity $\cos(x_i^h, \mathbf{d}^h)$ and $\cos(x_i^t, \mathbf{d}^t)$. To reduce the impact of redundant information in x_i , the context representation x_i is fed into the distillation module, where x_i is projected into the orthogonal space of irrelevant features to obtain the refined representation x_i^p . The score of context matching is $\cos(x_i^p, \mathbf{d})$. Finally, the overall matching score is the sum of entity and context matching scores.

3.2 Input-Description Encoding Module

The encoding module aims to encode the entity and context information in the input and label description into fixed-length representations for subsequent fine-grained matching.

3.2.1 Relation Description Encoder

Each candidate relation corresponds to a natural language label description $d \in \{d_{r_i^s} | i = 1, \dots, n\}$. For example, the relation *headquartered_in* corresponds to the description “the headquarters of an organization is located in a place”, and its encoded representation \mathbf{d} can be used for contextual matching. However, how an entity description is constructed (based on d), is important for a high-quality entity representation. In this subsection, we explore different ways to automatically construct and enrich entity description as follows:

Keyword: The entity hypernym (i.e. entity type) in d is directly used as the entity description. Take *headquartered_in* as an example, \mathbf{d}^h is “organization” and \mathbf{d}^t is “place”.

Synonyms: To further enrich the entity information, we use the words that mean exactly or nearly the same as the original hypernym extracted using Wikidata[†] and Thesaurus[‡]. Then, \mathbf{d}^h becomes

[†]<https://www.wikidata.org/>

[‡]<https://www.thesaurus.com/>

“organization, institution, company”.

Rule-based Template Filling: Inspired by prompt learning (Liu et al., 2021), a fluent and complete entity description may stimulate the pretrained model to output a high-quality entity representation. The synonym-extended hypernym sequence is filled into a template with some slots (i.e. *the head/tail entity types including [S], [S], ...*), and d^h becomes “*the head entity types including organization, institution, company*”. Empirical results show that expanding all candidate descriptions with the above template works well. The customized template design is not the focus of this paper, so we leave it to future work.

Following (Chen and Li, 2021), we adopt a fixed sentence-BERT as the implementation of the relation description encoder $f(\cdot)$, which encodes the above descriptions into fixed-length representations, that is $f(d) = \mathbf{d} \in \mathbb{R}^d$, $f(d^h) = \mathbf{d}^h \in \mathbb{R}^d$, $f(d^t) = \mathbf{d}^t \in \mathbb{R}^d$.

3.2.2 Input Instance Encoder

Given an input instance $x_i = \{w_1, w_2, \dots, w_n\}$, in which four reserved special tokens $[E_h], [\backslash E_h], [E_t], [\backslash E_t]$ are inserted to mark the beginning and end of the head entity e_{i1} and tail entity e_{i2} respectively. We obtain entity representation \mathbf{x}_i^h and \mathbf{x}_i^t by MaxPool the corresponding hidden states of entity tokens. Following (Baldini Soares et al., 2019), the context representation \mathbf{x}_i is obtained by concatenating the hidden states of special token $[E_h], [E_t]$.

$$\mathbf{h}_1, \dots, \mathbf{h}_n = \text{BERT}(w_1, \dots, w_n) \quad (1)$$

$$\mathbf{x}_i^h = \text{MaxPool}(\mathbf{h}_{b_h}, \dots, \mathbf{h}_{e_h}) \quad (2)$$

$$\mathbf{x}_i^t = \text{MaxPool}(\mathbf{h}_{b_t}, \dots, \mathbf{h}_{e_t}) \quad (3)$$

$$\mathbf{x}_i = \phi(\langle \mathbf{h}_{[E_h]} | \mathbf{h}_{[E_t]} \rangle), \quad (4)$$

where $\langle \cdot | \cdot \rangle$ denotes the concatenation operator. b_h, e_h, b_t, e_t denote the beginning and end position indexes of the head and tail entities respectively. $\mathbf{h}_{[E_h]}$ and $\mathbf{h}_{[E_t]}$ represent the hidden states of $[E_h]$ and $[E_t]$ respectively. Their corresponding position indexes are $b_h - 1$ and $b_t - 1$. $\phi(\cdot)$ denotes a linear layer with tanh activation, converting the dimension of $\langle \mathbf{h}_{[E_h]} | \mathbf{h}_{[E_t]} \rangle$ from $2n$ back to n .

3.3 Contextual Relation Feature Distillation

Due to the lack of explicit annotations to the relation-irrelevant components, it is non-trivial for the model to learn to identify them. This section

introduces how to reduce the negative impact of the redundant components on context matching.

3.3.1 Relation-Irrelevant Feature Aggregator

Given the output $\mathbf{h}_1, \dots, \mathbf{h}_n$ of the input instance encoder, we aggregate the relation-irrelevant features through a trainable query code $\mathbf{q} \in \mathbb{R}^d$ as follows:

$$(\alpha_1, \dots, \alpha_n) = \text{Softmax}(\mathbf{q} \cdot \mathbf{h}_1, \dots, \mathbf{q} \cdot \mathbf{h}_n) \quad (5)$$

$$\mathbf{x}_i^* = \sum_{j=1}^n \alpha_j \cdot \mathbf{h}_j, \quad (6)$$

This leads to an immediate question, **how do we make query code \mathbf{q} select relation-irrelevant features from context?** Intuitively, it is impossible for a relational classifier to discriminate the relations of input instances based on relation-irrelevant features. Therefore, we introduce a Gradient Reverse Layer (GRL) (Ganin and Lempitsky, 2015) and optimize \mathbf{q} by fooling the relational classifier.

$$\text{prob}_i = \text{Softmax}(\text{GRL}(\mathbf{x}_i^*) \cdot W + b) \quad (7)$$

$$\mathcal{L}_{ce,i} = \text{CrossEntropy}(y_i, \text{prob}_i), \quad (8)$$

where W and b are the weights and biases of the relation classifier. \mathbf{x}_i^* goes through a GRL layer before being fed into the classifier. GRL does not affect forward propagation but changes the gradient sign during backpropagation by multiplying $-\lambda$. That is, as the training proceeds, the classifier is optimized by gradient descent to reduce $\mathcal{L}_{ce,i}$, while the query code \mathbf{q} is optimized by gradient ascent to increase $\mathcal{L}_{ce,i}$, until no relational features are included in \mathbf{x}_i^* . The effectiveness of GRL has been verified in many literatures in the past (Ganin et al., 2016; Zhang et al., 2019).

3.3.2 Relation Feature Distillation Layer

The distillation module aims to reduce the negative impact of relation-irrelevant components on its representation and thus improving context matching. Given a context representation \mathbf{x}_i (refer to eq.4), as well as relation-irrelevant features \mathbf{x}_i^* (refer to eq.6), we achieve the above goal by projecting \mathbf{x}_i to the orthogonal space of \mathbf{x}_i^* . Specifically, we first project \mathbf{x}_i to the direction of \mathbf{x}_i^* to find the relation-irrelevant features $\hat{\mathbf{x}}_i$ mixed in \mathbf{x}_i as follows:

$$\hat{\mathbf{x}}_i = \text{Proj}(\mathbf{x}_i, \mathbf{x}_i^*) \quad (9)$$

$$\text{Proj}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{b}|} \cdot \frac{\mathbf{b}}{|\mathbf{b}|}, \quad (10)$$

where $\text{Proj}(\cdot, \cdot)$ denotes the projection operator and \mathbf{a}, \mathbf{b} are the input vectors of Proj . Then, we obtain the refined context representation \mathbf{x}_i^p by removing $\hat{\mathbf{x}}_i$ from \mathbf{x}_i as follows:

$$\mathbf{x}_i^p = \mathbf{x}_i - \hat{\mathbf{x}}_i. \quad (11)$$

Essentially, eqs. 9-11 project \mathbf{x}_i in the orthogonal direction of \mathbf{x}_i^* . The above process is illustrated in the right side of fig. 2.

3.4 Fine-Grained Matching and Training

Following the matching pattern of relational data, we decompose sentence-level matching into entity matching and context matching. For an input x_i and a candidate relation description $d \in \{d_{r_i^s} | i = 1, \dots, n\}$, the entity and context information are encoded into fixed-length representations $(\mathbf{x}_i^h, \mathbf{x}_i^t, \mathbf{x}_i^p)$, and $(\mathbf{d}^h, \mathbf{d}^t, \mathbf{d})$, respectively. The matching score between x_i and d is the sum of entity and context matching scores as follows:

$$s(x_i, d) = \alpha \cdot \cos(\mathbf{x}_i^h, \mathbf{d}^h) + \alpha \cdot \cos(\mathbf{x}_i^t, \mathbf{d}^t) + (1 - 2 \cdot \alpha) \cdot \cos(\mathbf{x}_i^p, \mathbf{d}), \quad (12)$$

where α is a hyper-parameter and $\cos(\cdot, \cdot)$ denotes the cosine operator. In order to optimize the above matching model and avoid over-fitting, we use margin loss as the objective function.

$$\delta_i = s(x_i, d_{y_i}) - \max_{j \neq y_i} (s(x_i, d_j)) \quad (13)$$

$$\mathcal{L}_{m,i} = \max(0, \gamma - \delta_i), \quad (14)$$

where $\gamma > 0$ is a hyper-parameter, meaning that the matching score of a positive pair must be higher than the closest negative pair. The overall training objective is as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N (\mathcal{L}_{ce,i} + \mathcal{L}_{m,i}), \quad (15)$$

where N is the batch size. When testing, the learned model is transferred to recognize the unseen relations in \mathcal{R}_u , by a match between the input and the descriptions of the unseen relations.

4 Experimental Setup

4.1 Datasets

FewRel (Han et al., 2018) is a few-shot relation classification dataset collected from Wikipedia and further hand-annotated by crowd workers, which contains 80 relations and 700 sentences in each

type of relation. **Wiki-ZSL** (Chen and Li, 2021) is derived from Wikidata Knowledge Base and consists of 93,383 sentences on 113 relation types. Compared with the FewRel dataset, Wiki-ZSL has more abundant relational information but inevitably has more noise in raw data since it is generated by distant supervision.

Following Chia et al. (2022), we randomly select 5 relations for validation set, $m \in \{5, 10, 15\}$ relations as unseen relations for testing set, and consider the remaining relations as seen relations for training set. Meanwhile, we randomly repeat the class selection process 5 times to ensure the reliability of the experiment results. We report the average results across different selections.

4.2 Compared Methods

To evaluate the effectiveness of our method, we make comparisons with a classic supervised method and state-of-the-art matching-based ZeroRE methods. We also compare a recent competitive seq2seq-based ZeroRE method.

R-BERT (Wu and He, 2019b). A SOTA supervised RE method. Following Chen and Li (2021), we adapt it to zero-shot setting by using the sentence representation to perform nearest neighbor search and generate zero-shot prediction.

ESIM (Levy et al., 2017). A classical matching-based ZeroRE method, which uses Bi-LSTM to encode the input and label description.

ZS-BERT (Chen and Li, 2021). A SOTA siamese-based ZeroRE method, which adopts BERT as the encoder to separately encode the input and relation description. In addition to classification loss, a metric-based loss is used to optimize representation space to improve nearest neighbor search.

PromptMatch (Sainz et al., 2021). A SOTA full encoding-based ZeroRE method, which adopts BERT to encode the concatenation of input pairs and model their fine-grained semantic interaction.

REPrompt (Chia et al., 2022). This baseline is a competitive seq2seq-based ZeroRE method. It uses GPT-2 to generate pseudo data of these relations to finetune the model. We use NoGen to denote the results without data augmentation.

4.3 Implementation Details

We use *Bert-base-uncased* as the input instance encoder, and we adopt a fixed sentence-Bert(Reimers and Gurevych, 2019) *stsb-bert-base* as the relation description encoder. We set AdamW(Loshchilov and Hutter, 2017) as the optimizer, and $2e - 6$ as

Unseen Labels	Method	Wiki-ZSL			FewRel		
		Prec.	Rec.	F_1	Prec.	Rec.	F_1
$m = 5$	R-BERT (Wu and He, 2019b)	39.22	43.27	41.15	42.19	48.61	45.17
	ESIM (Levy et al., 2017)	48.58	47.74	48.16	56.27	58.44	57.33
	ZS-BERT (Chen and Li, 2021)	71.54	72.39	71.96	76.96	78.86	77.90
	PromptMatch (Sainz et al., 2021)	77.39	75.90	76.63	91.14	90.86	91.00
	REPrompt(NoGen) (Chia et al., 2022)	51.78	46.76	48.93	72.36	58.61	64.57
	REPrompt (Chia et al., 2022)	70.66	83.75	76.63	90.15	88.50	89.30
	RE-Matching	78.19	78.41	78.30	92.82	92.34	92.58
$m = 10$	R-BERT (Wu and He, 2019b)	26.18	29.69	27.82	25.52	33.02	28.20
	ESIM (Levy et al., 2017)	44.12	45.46	44.78	42.89	44.17	43.52
	ZS-BERT (Chen and Li, 2021)	60.51	60.98	60.74	56.92	57.59	57.25
	PromptMatch (Sainz et al., 2021)	71.86	71.14	71.50	83.05	82.55	82.80
	REPrompt(NoGen) (Chia et al., 2022)	54.87	36.52	43.80	66.47	48.28	55.61
	REPrompt (Chia et al., 2022)	68.51	74.76	71.50	80.33	79.62	79.96
	RE-Matching	74.39	73.54	73.96	83.21	82.64	82.93
$m = 15$	R-BERT (Wu and He, 2019b)	17.31	18.82	18.03	16.95	19.37	18.08
	ESIM (Levy et al., 2017)	27.31	29.62	28.42	29.15	31.59	30.32
	ZS-BERT (Chen and Li, 2021)	34.12	34.38	34.25	35.54	38.19	36.82
	PromptMatch (Sainz et al., 2021)	62.13	61.76	61.95	72.83	72.10	72.46
	REPrompt(NoGen) (Chia et al., 2022)	54.45	29.43	37.45	66.49	40.05	49.38
	REPrompt (Chia et al., 2022)	63.69	67.93	65.74	74.33	72.51	73.40
	RE-Matching	67.31	67.33	67.32	73.80	73.52	73.66

Table 1: Main results on two relation extraction datasets. We report the average results of five runs and the improvement is significant (using a Wilcoxon signed-rank test; $p < 0.05$).

the learning rate. Based on the validation set, we conduct hyper-parameter selection. α is selected among $\{0.2, 0.33, 0.4\}$ and λ is selected among $\{0.1, 0.3, 0.5, 0.7\}$. Finally, we set $\alpha = 0.33$ and $\lambda = 0.5$ for all datasets. The batch size is set to 128. All experiments are conducted using an NVIDIA GeForce RTX 3090.

5 Results and Analysis

5.1 Main Results

The results on Wiki-ZSL and FewRel datasets are reported in tab. 1, which shows that the proposed method consistently outperforms previous SOTA methods when targeting at a different number of unseen relations. Specifically, simple classification loss only focuses on the discrimination between known relations, so the supervised method such as R-BERT fails on ZeroRE. Although ZS-BERT is designed for ZeroRE, the siamese scheme limits the word-level interaction between the input and relation description, leading to suboptimal performance. By contrast, our method compensates for the shortcoming by explicit modeling of fine-grained matching patterns of relational

data, thereby outperforming ZS-BERT by a large margin. Although full encoding scheme such as PromptMatch can implicitly model fine-grained interactions through self-attention, the proposed method is still able to outperform it. One possible reason is that the relational matching pattern, as an inductive bias, alleviates the over-fitting to seen relations in the training set and thus our model has better generalization. Compared with REPrompt, which is a seq2seq-based method, our method achieves better results without using pseudo data of new relations to fine-tune the model, showing the superiority of the method.

5.2 Ablation Study

To study the contribution of each component in the proposed method, we conduct ablation experiments on the two datasets and display the results in tab. 2. The results show that the matching performance is declined if the context distillation module is removed (i.e., w/o Proj), indicating that the relation-irrelevant information in the context disturbs the match of relational data, and projection in the distillation module is effective for reducing this impact. It is worth noting that entity information

Dataset	Method	Prec.	Rec.	F_1
Wiki-ZSL	w/o Proj.	66.13	67.18	66.65
	w/o Ent.	41.81	40.46	41.12
	w/o both	36.34	36.12	36.23
	Ours	67.31	67.33	67.32
FewRel	w/o Proj.	72.35	71.24	71.79
	w/o Ent.	49.16	41.51	45.01
	w/o both	37.20	32.43	34.65
	Ours	73.80	73.52	73.66

Table 2: Ablation study of our method ($m = 15$).

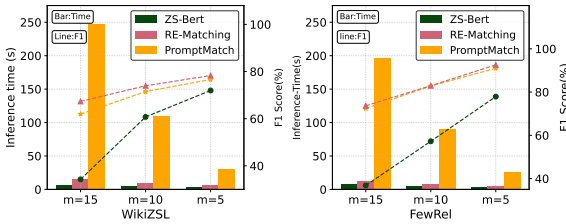


Figure 3: Comparison in terms of time consumption (Bars) and matching F_1 (Dotted lines). The fine-grained matching (pink) improves F_1 while maintaining the efficiency of the siamese scheme (i.e., ZS-BERT).

plays an important role in relational data (see w/o Ent). Explicitly modeling the matching between entities and their hypernyms significantly improves the performance of the model. As two vital components, when both context distillation and entity matching are removed (i.e., w/o both), the matching degenerates into sentence-level matching and the performance will be seriously hurt.

5.3 Efficiency Advantage

Fig. 3 shows the time consumption and matching F_1 scores on Wiki-ZSL and FewRel datasets. Take FewRel as an example, each relation contains 700 testing inputs. The siamese scheme (ZS-Bert and our RE-Matching) separately encodes input and descriptions and the encoding is run ($700 \cdot m + m$) times. By contrast, the full encoding scheme (PromptMatch) requires the concatenation of the text pair to be fed and the encoding is run ($700 \cdot m^2$) times. Clearly, as the number of new relations m increases, ZS-Bert and our RE-Matching have a significant efficiency advantage over PromptMatch that adopts a full encoding scheme. When $m = 15$, the inference time can be reduced by more than 10 times. Although our method takes slightly more time than ZS-BERT, the fine-grained matching brings a significant F_1 improvement. As shown

Model	Wiki-ZSL	FewRel
	FullEncoding→Ours (change)	FullEncoding→Ours (change)
BERT	61.95 → 67.32(5.37 ↑)	72.46 → 73.66(1.20 ↑)
RoBERTa	62.58 → 72.86(10.28 ↑)	73.79 → 73.82(0.03 ↑)
DistilBERT	57.05 → 67.41(10.36 ↑)	66.34 → 69.35(3.01 ↑)
DistilRoBERTa	57.46 → 68.14(10.68 ↑)	68.13 → 70.73(2.60 ↑)

Table 3: F_1 scores on two datasets. We compare the fine-grained matching with a full encoding scheme by varying the backbone to show consistency. Numbers in **bold** indicate whether the change is significant (using a Wilcoxon signed-rank test; $p < 0.05$).

in tab. 1, when $m = 15$, our method improves the F_1 score by 33.07% and 36.84% on two datasets respectively, compared with ZS-BERT.

5.4 Consistency over Various Encoders

In this section, we evaluate the effectiveness of our method by varying the selection of encoders. Tab. 3 shows the comparison results between ours and the full encoding scheme (i.e., PromptMatch) when $m = 15$. It can be observed that our method achieves consistent improvement. PromptMatch is able to learn the word-level interaction by self-attention on the input-description pair. However, the data-driven optimization paradigm usually learns spurious correlations in data pairs, especially in noisy data. By contrast, the matching pattern can be seen as a reasonable inductive bias. Modeling the pattern can reduce the overfitting risk. Therefore, our method consistently outperforms PromptMatch, especially on the noisy Wiki-ZSL (A distantly supervised dataset. See sec. 4.1).

5.5 Error Analysis

What errors of baselines our method is able to correct? (1) Our method reduces the negative impact of irrelevant components on context matching by projection. To intuitively show this, we use the attribution technique (Feng et al., 2018) to find words that are highly influential to the context matching score. A visualization case is shown in fig. 4. When using the projection, the model pays more attention to entities and relational phrases instead of irrelevant components (e.g., held, first, a) to make the prediction. (2) The entity matching score can provide more information to distinguish confusing relations. Taking P59:constellation_of_celestial_sphere as an example, its F_1 is only 0.123 without entity matching score. In this example, 79.71% of the incorrect cases are recognized as P361:part_of, due to the fact that the descriptions of P361 and P59



Figure 4: Visualization of the impact of words on context matching score. Darker colors indicate greater impact. The upper/lower sub-figure shows the results with/without projection respectively. The yellow and red boxes identify the head and tail entities respectively, while the blue box identifies the relational phrases [Relation: Winner].

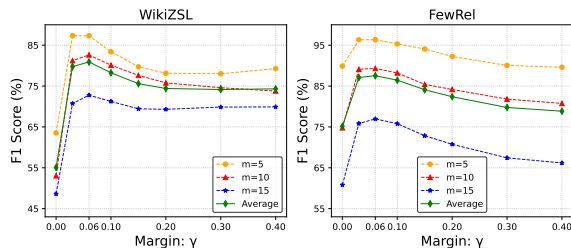


Figure 5: Model performance with different γ

are similar (a constellation is described as part of a celestial sphere). With entity matching, the type of head and tail entities is constrained and the F_1 is improved from 0.123 to 0.950.

What errors need to be addressed in further studies? Taking P937:place_of_work and P19:place_of_birth as examples, the entity pair type of both relations are person-location. Therefore, explicitly modeling entity matching does not lead to further improvement, when compared with the baselines. In addition, some abstract relations are difficult to accurately recognize. Take P460:said_to_be_the_same as an example. Such abstract relations do not have explicit entity types, and it is difficult to give a high-quality relation description. Therefore, the F_1 score of P460 is only 0.03.

5.6 Hyper-parameter Analysis

γ is an important hyper-parameter in our optimization objective. It means that the matching score of the positive pair should be at least γ higher than that of the negative pairs. In this subsection, we conduct experiments on two datasets (single class selection process) to study the influence of the value γ on matching performance. From fig. 5 we can obtain the following observation. First, as γ increases from 0, the model learns the difference between positive and negative pairs, thus assigning higher matching scores to input and correct description. When γ increases to a critical value (i.e., 0.06), the performance begins to decline gradually. This indicates that a too-large value makes the model overfit to known relation in training set, and then

Dataset	Method	Prec.	Rec.	F_1
Wiki-ZSL	Keyword	62.03	59.12	60.54
	Synonyms	65.28	63.53	64.39
	Template	67.31	67.33	67.32
FewRel	Keyword	72.01	71.89	71.91
	Synonyms	72.24	71.68	71.96
	Template	73.80	73.52	73.66

Table 4: Comparison of different construction methods of entity descriptions ($m=15$).

lose generalization. Finally, even if γ is increased to a very large value, the matching does not crash. This shows that our method has good robustness.

5.7 The Influence of Entity Description

In a description, there are usually only one or two words that identify the entity type. As shown in tab. 4, we explore how to build a high-quality entity representation based on the words for entity matching. A simple way is to directly encode the words as entity representation. However, insufficient entity information limits matching performance. We further use synonyms to enrich entity information and improve the F_1 score by 3.85% in Wiki-ZSL. In order to further construct a complete and fluent entity description, we fill the synonym-extended word sequence into the template slot. Compared with the original *keyword* method, the two operations improve F_1 score by 6.78% and 1.75% on the two datasets respectively.

6 Conclusions

In this work, we propose a fine-grained semantic matching method for ZeroRE. This method explicitly models the matching pattern of relational data, by decomposing the similarity score into entity and context matching scores. We explore various ways to enrich entity description and thus facilitating high-quality entity representation. The context distillation module effectively reduces the negative impact of irrelevant components on

context matching. Experimental results show that our method achieves higher matching F_1 score and has an inference speed 10 times faster when compared with SOTA methods.

Limitations

Elaborated relation descriptions are the foundation of the matching-based methods to achieve superior performance. Although we have proposed some ways to enrich the entity information in the descriptions, it is still a promising direction to explore more diversified and effective ways to enrich relation description (e.g. ensemble of multiple descriptions). We leave this as our future work.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No.61976056,62076069,62206057), Shanghai Rising-Star Program (23QA1400200), and Natural Science Foundation of Shanghai (23ZR1403500).

References

- Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiro Sadamasa, Yanjun Qi, Olivier Chapelle, and Kilian Weinberger. 2009. [Supervised semantic indexing](#). In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, page 187–196, New York, NY, USA. Association for Computing Machinery.
- Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. [Matching the blanks: Distributional similarity for relation learning](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.
- Chih-Yao Chen and Cheng-Te Li. 2021. [ZS-BERT: Towards zero-shot relation extraction with attribute representation learning](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3470–3479, Online. Association for Computational Linguistics.
- Yew Ken Chia, Lidong Bing, Soujanya Poria, and Luo Si. 2022. [RelationPrompt: Leveraging prompts to generate synthetic data for zero-shot relation triplet extraction](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 45–57, Dublin, Ireland. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. [Pathologies of neural models make interpretations difficult](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.
- Yaroslav Ganin and Victor Lempitsky. 2015. [Unsupervised domain adaptation by backpropagation](#). In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France. PMLR.
- Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030.
- Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. [More data, more relations, more context and more openness: A review and outlook for relation extraction](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 745–758, Suzhou, China. Association for Computational Linguistics.
- Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. [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*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. [Learning deep structured semantic models for web search using clickthrough data](#). In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, CIKM '13, page 2333–2338, New York, NY, USA. Association for Computing Machinery.
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. [Zero-shot relation extraction via reading comprehension](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *CoRR*, abs/2107.13586.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Abiola Obamuyide and Andreas Vlachos. 2018. [Zero-shot relation classification as textual entailment](#). In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 72–78, Brussels, Belgium. Association for Computational Linguistics.
- N. Reimers and I. Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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)*.
- Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. [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*, pages 1199–1212, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. 2018. Starspace: Embed all the things! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Shanchan Wu and Yifan He. 2019a. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2361–2364.
- Shanchan Wu and Yifan He. 2019b. [Enriching pre-trained language model with entity information for relation classification](#). In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19*, page 2361–2364, New York, NY, USA. Association for Computing Machinery.
- Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. [Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505, Vancouver, Canada. Association for Computational Linguistics.
- Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W. Bruce Croft, Jun Huang, and Haiqing Chen. 2018. [Response ranking with deep matching networks and external knowledge in information-seeking conversation systems](#). In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '18*, page 245–254, New York, NY, USA. Association for Computing Machinery.
- Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. [From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing](#). In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18*, page 497–506, New York, NY, USA. Association for Computing Machinery.
- Kai Zhang, Hefu Zhang, Qi Liu, Hongke Zhao, Hengshu Zhu, and Enhong Chen. 2019. [Interactive attention transfer network for cross-domain sentiment classification](#). In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'19/IAAI'19/EAAI'19*. AAAI Press.
- Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. [Modeling multi-turn conversation with deep utterance aggregation](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3740–3752, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Jun Zhao, Tao Gui, Qi Zhang, and Yaqian Zhou. 2021. [A relation-oriented clustering method for open relation extraction](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9707–9718, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
Limitations section
- A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.
- A3. Do the abstract and introduction summarize the paper’s main claims?
abstract and introduction sections
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

section 3

- B1. Did you cite the creators of artifacts you used?
section 3
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
section 3
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
section 4

C Did you run computational experiments?

section 5

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

section 4

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

section 4

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

section 4

D Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

No response.

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

No response.

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

No response.

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

No response.

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

No response.