

# Fusion Makes Perfection: An Efficient Multi-Grained Matching Approach for Zero-Shot Relation Extraction

Shilong Li\*, Ge Bai\*, Zhang Zhang\*, Ying Liu  
Chenji Lu, Daichi Guo, Ruifang Liu, Yong Sun†

Beijing University of Posts and Telecommunications, China  
lishilong2019210645@bupt.edu.cn

## Abstract

Predicting unseen relations that cannot be observed during the training phase is a challenging task in relation extraction. Previous works have made progress by matching the semantics between input instances and label descriptions. However, fine-grained matching often requires laborious manual annotation, and rich interactions come with significant computational overhead. In this work, we propose an efficient multi-grained matching approach that uses virtual entity matching to reduce manual annotation cost, and fuses coarse-grained recall and fine-grained classification for rich interactions with guaranteed inference speed. Experimental results show that our approach outperforms the previous State Of The Art (SOTA) methods, and achieves a balance between inference efficiency and prediction accuracy in zero-shot relation extraction tasks. Our code is available at <https://github.com/longls777/EMMA>.

## 1 Introduction

Relation Extraction (RE) is an important task of Natural Language Processing (NLP), which aims to identify the relation between a pair of entities within a given sentence. Previous RE models (Han et al., 2020; Peng et al., 2020; Zhao et al., 2021) have impressive performance through large-scale supervised learning based on high-quality labeled data. However, collecting sufficient data for every new relation type is laborious in practice. This leads to the necessity of zero-shot RE task, which involves extracting unobserved relations.

Recently, semantic matching (Obamuyide and Vlachos, 2018) has become a mainstream paradigm of zero-shot RE, which matches a given input with a corresponding label description. PromptMatch (Sainz et al., 2021) performed self-attention over

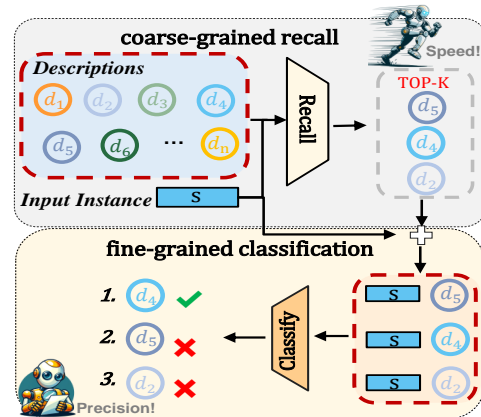


Figure 1: The overall process of our method. The coarse-grained recall refers to the rough and rapid screening of several possible results, while the fine-grained classification denotes the detailed discrimination of these possible results.

each instance-description pair to enrich interaction, but increased computational overhead. ZS-Bert (Chen and Li, 2021) enabled fast inference by encoding the input and description separately, and then storing and reusing the representation of descriptions for each input. However, the lack of interaction during the encoding also limits the performance of the model. RE-Matching (Zhao et al., 2023) introduced a unique fine-grained matching pattern and improved both the accuracy and speed by ignoring redundant components in the instance and matching the entities with their hypernyms in the description. However, this approach relies on manual annotation of entity hypernyms in label descriptions and still lacks the interaction between instances and descriptions. Therefore, how to achieve a balance between efficiency and accuracy without using additional labor costs is a pressing issue.

To address this issue, we propose an **Efficient Multi-Grained Matching Approach** (EMMA). In this work, we generate virtual entity representations of descriptions in semantic matching instead of annotating descriptions to avoid manual costs. Additionally, we utilize a fusion of coarse-grained

\*The first three authors contribute equally.

†Yong Sun is the corresponding author.

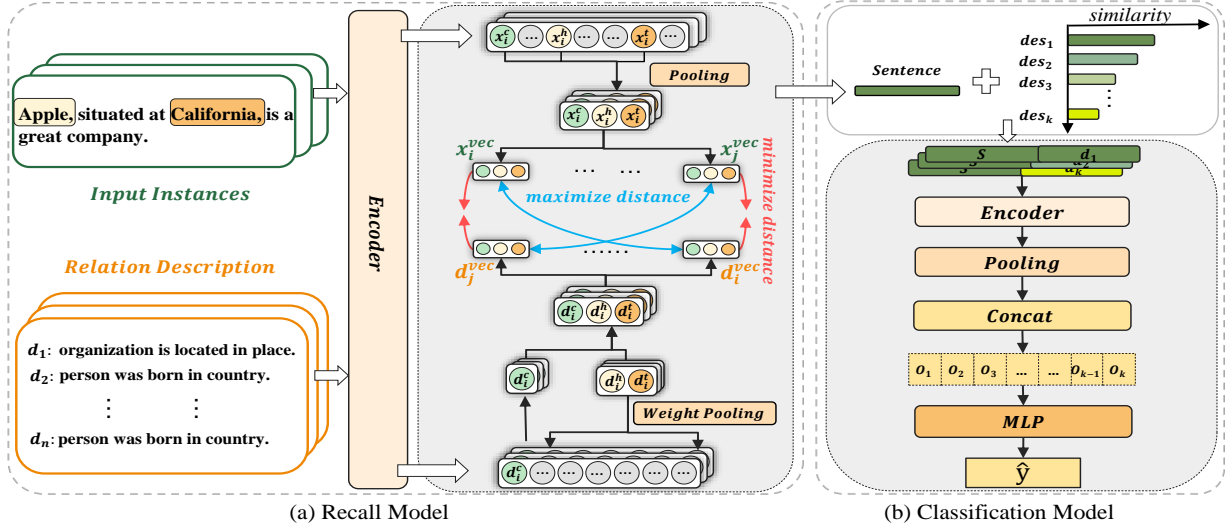


Figure 2: The overall architecture of EMMA. (a) The recall model swiftly matches to retrieve the top  $k$  most probable relations. (b) The classification model further distinguishes among these similar relations.

recall and fine-grained classification. Specifically, a coarse-grained filter is used to improve inference speed and select several candidate relations for each input, while a fine-grained classifier enhances instance-description interaction, enabling more accurate selection from relation candidates to improve prediction precision.

We summarize the contributions as follows:

- To the best of our knowledge, EMMA is the first work fusing the coarse-grained recall stage and fine-grained classification stage to achieve a balance of accuracy and inference speed.
- We introduce a virtual entity matching method to achieve effective semantic matching as well as avoid laborious manual annotation.
- Extensive experiments on different datasets and settings show EMMA outperforms previous SOTA methods, which demonstrates the efficiency and effectiveness of our approach.

## 2 Approach

### 2.1 Task Formulation

The zero-shot RE task is designed to learn from the seen relations  $\mathcal{R}_s = \{r_1^s, r_2^s, \dots, r_n^s\}$  to identify unseen relations  $\mathcal{R}_u = \{r_1^u, r_2^u, \dots, r_m^u\}$ . These two sets are disjoint, and the model only uses  $\mathcal{R}_s$  during the training phase. Similar to the previous work (Zhao et al., 2023; Chen and Li, 2021), we formulate zero-shot RE as a semantic matching

task. We further subdivide it into two stages: recall stage and classification stage.

In the recall stage, the training set comprises  $N$  samples  $\mathcal{D} = \{(x_i, e_i^h, e_i^t, y_i, d_i) | i = 1, \dots, N\}$ , where  $x_i$  is input instance,  $e_i^h$  is head entity,  $e_i^t$  is tail entity,  $y_i \in \mathcal{R}_s$  is corresponding relation and  $d_i$  is the relation description. We optimize a recall model  $\mathcal{M}_r(x, e^h, e^t, d) \rightarrow s \in \mathbb{R}$  on  $\mathcal{R}_s$ , where  $s$  represents the matching score between the instance and description. Then we recall top  $k$  relation exhibiting the highest matching scores.

In the classification stage, for the instance and the top  $k$  relation descriptions, we optimize a fine-grained classification model  $\mathcal{M}_c(x, e^h, e^t, d_1, d_2, \dots, d_k) \rightarrow \hat{y}$ , where  $\hat{y}$  is the predicted probability.

During testing on  $\mathcal{R}_u$ , given a sample  $(x_u, e_u^h, e_u^t)$ , we use  $\mathcal{M}_r$  to obtain the top  $k$  most probable relation at a coarse-grained level, and use  $\mathcal{M}_c$  to further distinguish these relations at a fine-grained level, obtaining the most probable one.

### 2.2 Coarse-grained recall

To rapidly query the relation corresponding to the input instance without tediously encoding and matching each pair (Sainz et al., 2021), we adopt a dual-tower-like architecture (Yi et al., 2019), which allows for precomputing representations of numerous relations to facilitates swift matching.

#### 2.2.1 Input Instance Encoder

Given an input instance  $x = \{w_1^x, \dots, w_n^x\}$ , distinct special tokens  $[E_h]$ ,  $[\backslash E_h]$ ,  $[E_t]$ ,  $[\backslash E_t]$  are

employed to wrap the head entity and tail entity, respectively. After inputting  $x$  into a pre-trained encoder, we utilize the last hidden states of special tokens  $[E_h]$ ,  $[E_t]$ , and  $[CLS]$  (refer to  $w_0^x$ ) as representations of head entity, tail entity, and contextual information, which is formulated as follows:

$$h_0^x, h_1^x, \dots, h_n^x = \mathbf{BERT}(w_0^x, w_1^x, \dots, w_n^x) \quad (1)$$

$$x^c = h_0^x, x^h = h_{[E_h]}^x, x^t = h_{[E_t]}^x \quad (2)$$

Then we combine the representation of head entity  $x^h \in \mathbb{R}^d$ , tail entity  $x^t \in \mathbb{R}^d$ , and the contextual information  $x^c \in \mathbb{R}^d$  to form the comprehensive representation  $x^{vec} \in \mathbb{R}^{3d}$  of the input instance.

$$x^{vec} = x^c \oplus x^h \oplus x^t \quad (3)$$

where  $d$  is the hidden dimension of the encoder and  $\oplus$  denotes the concatenation operator.

### 2.2.2 Virtual Entity Matching

Although the description of corresponding relation  $d = \{w_1^d, \dots, w_n^d\}$  is easily obtainable (e.g. from Wikipedia), manually annotating the entity hypernyms within various relations is still time-consuming and laborious. Therefore, we directly input relation descriptions into the pre-trained encoder. Then, we employ two weight pooling layers with different parameters to obtain separate virtual entity representations  $d^h \in \mathbb{R}^d$  and  $d^t \in \mathbb{R}^d$ . Similar to Section 2.2.1, we use the hidden states corresponding to the  $[CLS]$  token (refer to  $w_0^d$ ) as the contextual representation  $d^c \in \mathbb{R}^d$ , and concatenate these three to obtain the comprehensive representation  $d^{vec} \in \mathbb{R}^{3d}$  of the relation description.

$$h_0^d, h_1^d, \dots, h_n^d = \mathbf{BERT}(w_0^d, w_1^d, \dots, w_n^d) \quad (4)$$

$$d^c = h_0^d \quad (5)$$

$$d^h = \mathit{WeightPooling}_1(h_1^d, \dots, h_n^d) \quad (6)$$

$$d^t = \mathit{WeightPooling}_2(h_1^d, \dots, h_n^d) \quad (7)$$

$$d^{vec} = d^c \oplus d^h \oplus d^t \quad (8)$$

For the weight pooling, we employ the scheme proposed by Lin et al. (2017), utilizing an attention mechanism over the last hidden states of the pre-trained encoder to generate representations of virtual entities, which is formulated as follows:

$$H = (h_1^d, \dots, h_n^d) \quad (9)$$

$$A = \mathit{softmax}(HW + b) \quad (10)$$

$$d^* = AH \quad (11)$$

where  $H \in \mathbb{R}^{(L-1) \times d}$  is the last hidden states of the encoder excluding  $[CLS]$  token ( $L$  denotes the max sequence length).  $W$  is a linear layer of  $(L-1) \times 1$ ,  $b \in \mathbb{R}^{L-1}$  is the bias, and  $A \in \mathbb{R}^{L-1}$  denotes the final weights. The final representation  $d^* \in \mathbb{R}^d$  is obtained by weighting  $H$  using  $A$ .

### 2.2.3 Contrastive Learning

When  $N$  input instances  $\{x_1, \dots, x_N\}$  and their corresponding relation descriptions  $\{d_1, \dots, d_N\}$  are input into the encoder within a mini-batch, we obtain the representations of instance  $x_i^{vec}$  and description  $d_i^{vec}$ ,  $i \in [1, N]$ . To effectively learn the matching relationship between  $x_i^{vec}$  and  $d_i^{vec}$ , we utilize a contrastive learning method, where  $d_i^{vec}$  serves as a positive sample and other  $N-1$  samples within the mini-batch  $d_j^{vec}$  ( $j \neq i$ ) serve as negative samples. The goal of contrastive learning is to minimize the distance between  $x_i^{vec}$  and  $d_i^{vec}$  while maximizing the distance from  $d_j^{vec}$ .

We utilize cosine similarity as the measurement and employ the infoNCE(van den Oord et al., 2018) as the contrastive loss function:

$$\mathcal{L}_i = -\log \frac{e^{\text{sim}(x_i^{vec}, d_i^{vec})/\tau}}{\sum_{j=1}^N e^{\text{sim}(x_i^{vec}, d_j^{vec})/\tau}} \quad (12)$$

where  $\tau$  is a temperature hyperparameter and  $\text{sim}(\cdot)$  is the cosine similarity.

### 2.3 Fine-grained classification

In the recall stage, we obtain representations of input instances and relation descriptions separately for quick query matching. However, the lack of interaction between the instances and descriptions limits the model's performance ceiling. To tackle this issue, we propose fine-grained classification after coarse-grained recall, which jointly encodes instances and descriptions.

In the classification stage, during training, for each input instance  $x$ ,  $k$  relation descriptions  $D = \{d_1, \dots, d_k\}$  are selected from the mini-batch of the recall stage, which includes  $d_+$  corresponding to the entity relation of  $x$ , and top  $k-1$  descriptions with the highest matching scores excluding  $d_+$ . The objective of classification is to select  $d_+$  from  $D$ . We formulate this process as follows:

$$O_i = \mathit{Pooling}(\mathbf{BERT}(\langle x \oplus d_j \rangle)) \quad (13)$$

$$\hat{y} = \mathit{MLP}(O_0 \oplus O_1 \oplus \dots \oplus O_k) \quad (14)$$

Unseen Labels	Method	Wiki-ZSL			FewRel		
		Prec.	Rec.	$F_1$	Prec.	Rec.	$F_1$
m=5	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(Chia et al., 2022)	70.66	83.75	76.63	90.15	88.50	89.30
	RE-Matching(Zhao et al., 2023)	78.19	78.41	78.30	92.82	92.34	92.58
	EMMA(onlyRecall)	89.30	90.10	89.70	93.68	92.76	93.22
	<b>EMMA</b>	<b>91.32</b>	<b>90.65</b>	<b>90.98</b>	<b>94.87</b>	<b>94.48</b>	<b>94.67</b>
m=10	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(Chia et al., 2022)	68.51	74.76	71.50	80.33	79.62	79.96
	RE-Matching(Zhao et al., 2023)	74.39	73.54	73.96	83.21	82.64	82.93
	EMMA(onlyRecall)	85.99	84.37	85.17	86.67	84.32	85.48
	<b>EMMA</b>	<b>86.00</b>	<b>84.55</b>	<b>85.27</b>	<b>87.97</b>	<b>86.48</b>	<b>87.22</b>
m=15	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(Chia et al., 2022)	63.69	67.93	65.74	74.33	72.51	73.40
	RE-Matching(Zhao et al., 2023)	67.31	67.33	67.32	73.80	73.52	73.66
	EMMA(onlyRecall)	76.83	75.79	76.31	78.24	75.77	76.99
	<b>EMMA</b>	<b>78.51</b>	<b>77.63</b>	<b>78.07</b>	<b>80.47</b>	<b>79.73</b>	<b>80.10</b>

Table 1: Main results on Wiki-ZSL and FewRel dataset. We report the average results obtained from running with five random seeds ( $k = 2$ ) and the improvement is significant (using a Wilcoxon signed-rank test;  $p < 0.05$ ).

$$\mathcal{L}_c = -\log\left(\frac{e^{\hat{y}_+}}{\sum_{i=1}^k e^{\hat{y}_i}}\right) \quad (15)$$

where  $O_i$  is the representation of instance-description pair obtained by extracting the last hidden state of the  $[CLS]$  token and  $\hat{y}$  is the predicted probability. We utilize cross-entropy as the loss function for classification.

During testing, the top  $k$  descriptions with the highest matching scores are selected as input.

### 3 Experiments

We conduct our experiments on the FewRel (Han et al., 2018) and Wiki-ZSL (Chen and Li, 2021) datasets. Specific details about the datasets and experimental details are provided in appendix A.

#### 3.1 Main results

Table 1 displays the experimental results on the Wiki-ZSL and FewRel datasets, showing that our proposed method significantly outperforms the previous SOTA results by a large margin when predicting different numbers of unseen relations, specifically when  $m = 15$ , it achieves at least a 11% improvement in F1 scores on Wiki-ZSL and a 6% improvement on FewRel. Even the EMMA model without the classification (onlyRecall), which selects the relation with the highest

Dataset	Method	Prec.	Rec.	$F_1$
FewRel	w/o Vir.	76.43	76.02	76.22
	w/o Cla.	78.24	75.77	76.99
	w/o both	75.54	75.12	75.33
	<b>Ours</b>	<b>80.47</b>	<b>79.73</b>	<b>80.10</b>

Table 2: Ablation study on FewRel ( $m = 15, k = 2$ ).

matching score as the prediction result, still outperforms the SOTA model. Moreover, compared to RE-Matching, EMMA employs virtual entity matching, avoiding the human effort required for annotated descriptions. Upon integrating the classification model, the complete version of EMMA extensively augments the interaction between the input sentence and relation description, further boosting the model’s performance. These showcase the superiority of our model.

#### 3.2 Ablation Study

Table 2 presents the results of ablation experiments, which indicates that removing the virtual entity matching (w/o Virt.) and the classification (w/o Cla.) individually both result in decreased model performance. This illustrates the effectiveness of virtual entity matching during the recall stage and



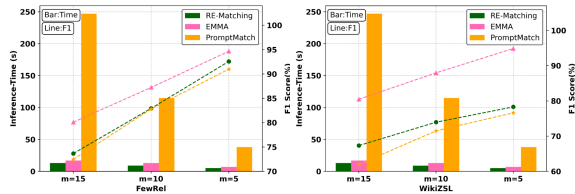


Figure 3: Comparison in terms of runtime (Bars) and matching F1 (Dotted lines).

the efficacy of the classification model designed to enhance interaction for identifying similar relations. When both are removed (w/o both), the model degrades to a simple semantic matching model, leading to a significant decline in performance.

### 3.3 Inference Efficiency

Figure 3 shows the inference runtime and matching F1 scores. As the number of new relations  $m$  increases, EMMA proves more efficient than PromptMatch. While it takes slightly longer than RE-Matching, EMMA significantly improves F1 scores. Detailed analysis is in appendix F.

## 4 Conclusions

In this work, we propose a fusion method for ZeroRE named EMMA, which enhances performance in the ZeroRE task by combining coarse-grained recall and fine-grained classification, while maintaining efficient inference capabilities. Experimental results demonstrate that our approach outperforms SOTA methods in matching F1 scores while maintaining rapid inference.

## 5 Limitations

Our proposed method has only been experimented on zero-shot relation extraction tasks and has not been applied in other domains of information extraction, such as named entity recognition. However, the underlying principles embedded within EMMA might potentially be generalized and applied to other related tasks.

## References

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.

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.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. [A structured self-attentive sentence embedding](#). *ArXiv*, abs/1703.03130.

Yury Malkov and Dmitry A. Yashunin. 2016. [Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:824–836.

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.

Hao Peng, Tianyu Gao, Xu Han, Yankai Lin, Peng Li, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2020. [Learning from context or names? an empirical study on neural relation extraction](#).

Oscar Sainz, Oier Lopez de Lacalle, Gorra 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.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. [Representation learning with contrastive predictive coding](#). *ArXiv*, abs/1807.03748.

Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao,

Li Wei, and Ed H. Chi. 2019. [Sampling-bias-corrected neural modeling for large corpus item recommendations](#). *Proceedings of the 13th ACM Conference on Recommender Systems*.

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.

Jun Zhao, WenYu Zhan, Xin Zhao, Qi Zhang, Tao Gui, Zhongyu Wei, Junzhe Wang, Minlong Peng, and Mingming Sun. 2023. [RE-matching: A fine-grained semantic matching method for zero-shot relation extraction](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6680–6691, Toronto, Canada. Association for Computational Linguistics.

## A Experimental setup

### A.1 Datasets

**FewRel** (Han et al., 2018) is a dataset designed for few-shot relation classification. It’s sourced from Wikipedia and involves manual annotation by crowd workers. It comprises 80 relations, each having 700 associated sentences. **Wiki-ZSL** (Chen and Li, 2021) originates from the Wikidata Knowledge Base, boasting 94,383 sentences spanning across 113 relation types. In Wiki-ZSL, entities are extracted from Wikipedia articles and linked to the Wikidata knowledge base. This method of remotely supervised generation results in Wiki-ZSL containing more noise than FewRel.

For the accuracy and comparability of experimental results, similar to Zhao et al. (2023), we randomly selected  $m \in \{5, 10, 15\}$  relations as the validation set,  $m$  relations as the test set, and the remainder as the training set. Simultaneously, we chose 5 different random seeds for dataset partitioning and experimentation, reporting the average results of these experiments.

### A.2 Implementation Details

We utilize *Bert-base-uncased* as the pre-trained encoder, which is then fine-tuned for our purposes. In the recall model, the encoder for the input sentence shares parameters with the encoder for the relation description. The encoder in the classification model has its separate parameters. The recall model and classification model are jointly trained in the experiment and we discuss the differences between joint training and separate training in appendix E.

The temperature  $\tau$  for the infoNCE loss is set to 0.02. We use AdamW optimizer with a learning rate of  $2e - 5$  and a batch size of 64. We train the model for 5 epochs with a warm-up of 100 steps. All experiments are conducted using an NVIDIA RTX A6000.

## B Ablation Experiments on Wiki-ZSL

Dataset	Method	Prec.	Rec.	$F_1$
Wiki-ZSL	w/o Virt.	74.03	74.74	74.38
	w/o Cla.	76.83	75.79	76.31
	w/o both	71.52	70.93	71.22
	<b>Ours</b>	<b>78.51</b>	<b>77.63</b>	<b>78.07</b>

Table 3: Ablation study on Wiki-ZSL ( $m = 15, k = 2$ ).

The ablation experiments conducted on the Wiki-ZSL dataset align with our conclusion that both virtual entity matching and classification components contribute beneficially to improving model performance.

## C Analysis of classification performance

Figure 4 illustrates an instance where the classification model corrects a recalled result. However, it’s possible for the top 1 result obtained by the recall model to be the correct one, yet after classification, an incorrect result is generated. Nonetheless, the experimental results in Table 1 comparing EMMA and EMMA (onlyRecall) indicate that the number of corrections by the classification model is greater than the number of errors corrected. This demonstrates the effectiveness of fine-grained classification.

## D Difference over various $k$

Figure 5 illustrates the change in EMMA’s F1 scores across different values of  $k$  on the FewRel and Wiki-ZSL datasets. As  $k$  increases (from 2 to 4), the model’s F1 score gradually decreases. This could be attributed to the increased difficulty in classification as the model needs to discern among a larger set of relations when  $k$  grows. How to mitigate this decline in such scenarios can be considered as a future research direction.

## E Differences between training methods

In joint training, we train the recall model and the classification model at the same time, which means

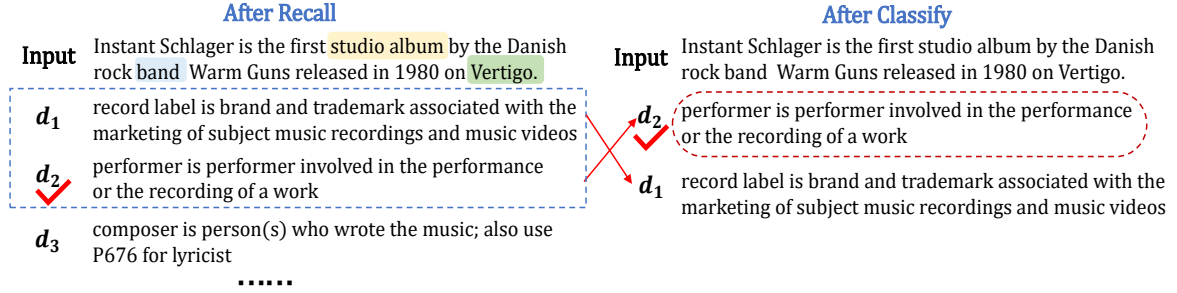


Figure 4: This is an example showcasing the role of the classification model. During the recall stage, the correct relation description wasn’t ranked first, yet through the fine-grained classification model’s correction, the accurate result was eventually obtained.

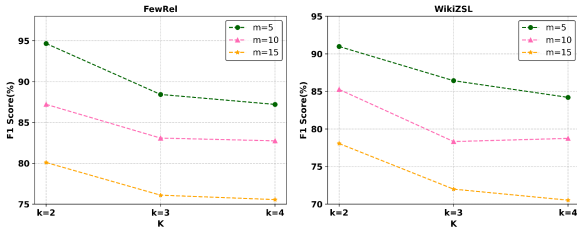


Figure 5: The F1 scores of EMMA across different values of  $k$ .

Training approach	Prec.	Rec.	$F_1$
joint training	<b>94.87</b>	<b>94.48</b>	<b>94.67</b>
separate training	94.62	94.36	94.49

Table 4: Experimental comparison on FewRel ( $m = 5$ ,  $k = 2$ ).

that the loss from the classification stage will back-propagate to the recall stage. In separate training, the recall model is trained first, and then the classification model is trained based on the output of the recall model. Regardless of the method, we ensure that the input to the classification model includes the correct relation.

From the experimental results, it can be observed that the difference between separate training and joint training is not significant.

## F Inference Efficiency Analysis

For both RE-Matching (Zhao et al., 2023) and EMMA, the representation vectors of relation descriptions can be pre-inferred. When inputting an instance, its obtained vector needs to be compared with each description vector individually. Assuming there are  $m$  instances and  $n$  relations, both models need to process this. The inference speed of RE-Matching should be  $O(m * n + n)$ , while EMMA, due to the inclusion of a fine-grained classification model, operates at  $O(m * n + m + n)$ .

However, in real-world scenarios where both  $m$  and  $n$  are large, the time complexity of both models tends toward  $O(m * n)$ , making the inference speed of EMMA and RE-Matching essentially similar. Certainly, we could use neighbor search methods like HNSW (Malkov and Yashunin, 2016) to reduce the time complexity of one-to-one matching in the recall stage. However, that is not the focus of this work.

Taking FewRel as an example, each relation comprises 700 test input instances. RE-Matching and our EMMA encode the input sentences and descriptions separately, with encoding performed  $(700 * n + n)$  times and  $(700 * n + 700 + n)$  times, respectively. In contrast, PromptMatch requires concatenation of text pairs for input and involves encoding performer  $(700 * n^2)$  times.