Learning from Context or Names? An Empirical Study on Neural Relation Extraction

Hao Peng[1*], **Tianyu Gao**[2*], **Xu Han**[1], **Yankai Lin**[3], **Peng Li**[3], **Zhiyuan Liu**[1†], **Maosong Sun**[1], **Jie Zhou**[3]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Princeton University, Princeton, NJ, USA
[3]Pattern Recognition Center, WeChat AI, Tencent Inc, China
{h-peng17,hanxu17}@mails.tsinghua.edu.cn, tianyug@princeton.edu

Abstract

Neural models have achieved remarkable success on relation extraction (RE) benchmarks. However, there is no clear understanding which type of information affects existing RE models to make decisions and how to further improve the performance of these models. To this end, we empirically study the effect of two main information sources in text: **textual context** and **entity mentions (names)**. We find that (i) while context is the main source to support the predictions, RE models also heavily rely on the information from entity mentions, most of which is type information, and (ii) existing datasets may leak shallow heuristics via entity mentions and thus contribute to the high performance on RE benchmarks. Based on the analyses, we propose an entity-masked contrastive pre-training framework for RE to gain a deeper understanding on both textual context and type information while avoiding rote memorization of entities or use of superficial cues in mentions. We carry out extensive experiments to support our views, and show that our framework can improve the effectiveness and robustness of neural models in different RE scenarios. All the code and datasets are released at https://github.com/thunlp/RE-Context-or-Names.

1 Introduction

Relation extraction (RE) aims at extracting relational facts between entities from text, e.g., extracting the fact (SpaceX, `founded by`, Elon Musk) from the sentence in Figure 1. Utilizing the structured knowledge captured by RE, we can construct or complete knowledge graphs (KGs), and eventually support downstream applications like question answering (Bordes et al., 2014), dialog systems (Madotto et al., 2018) and search engines (Xiong et al., 2017). With the recent advance of deep learning, neural relation extraction (NRE) models (Socher et al., 2012; Liu et al., 2013; Baldini Soares et al., 2019) have achieved the latest state-of-the-art results and some of them are even comparable with human performance on several public RE benchmarks.

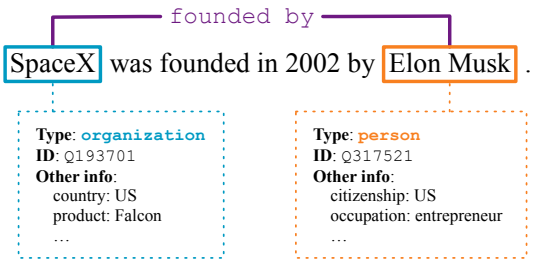

Figure 1: An example for the information provided by textual context and entity mentions in a typical RE scenario. From mentions, we can acquire type information and link entities to KGs, and access further knowledge about them. The IDs in the figure are from Wikidata.

The success of NRE models on current RE benchmarks makes us wonder *which type of information these models actually grasp to help them extract correct relations*. The analysis of this problem may indicate the nature of these models and reveal their remaining problems to be further explored. Generally, in a typical RE setting, there are two main sources of information in text that might help RE models classify relations: textual context and entity mentions (names).

From human intuition, **textual context** should be the main source of information for RE. Researchers have reached a consensus that there exist interpretable patterns in textual context that express relational facts. For example, in Figure 1, "... *be founded ... by ...*" is a pattern for the relation `founded by`. The early RE systems (Huffman, 1995; Califf and Mooney, 1997) formalize patterns into string templates and determine relations by

[*] Equal contribution
[†] Corresponding author e-mail: liuzy@tsinghua.edu.cn

matching these templates. The later neural models (Socher et al., 2012; Liu et al., 2013) prefer to encode patterns into distributed representations and then predict relations via representation matching. Compared with rigid string templates, distributed representations used in neural models are more generalized and perform better.

Besides, **entity mentions** also provide much information for relation classification. As shown in Figure 1, we can acquire the types of entities from their mentions, which could help to filter out those impossible relations. Besides, if these entities can be linked to KGs, models can introduce external knowledge from KGs to help RE (Zhang et al., 2019; Peters et al., 2019). Moreover, for pre-trained language models, which are widely adopted for recent RE models, there may be knowledge about entities inherently stored in their parameters after pre-training (Petroni et al., 2019).

In this paper, we carry out extensive experiments to study to what extent RE models rely on the two information sources. We find out that:

(1) Both context and entity mentions are crucial for RE. As shown in our experiments, while context is the main source to support classification, entity mentions also provide critical information, most of which is the type information of entities.

(2) Existing RE benchmarks may leak shallow cues via entity mentions, which contribute to the high performance of existing models. Our experiments show that models still can achieve high performance only given entity mentions as input, suggesting that there exist biased statistical cues from entity mentions in these datasets.

The above observations demonstrate how existing models work on RE datasets, and suggest a way to further improve RE models: we should enhance them via better understanding context and utilizing entity types, while preventing them from simply memorizing entities or exploiting biased cues in mentions. From these points, we investigate an entity-masked contrastive pre-training framework for RE. We use Wikidata to gather sentences that may express the same relations, and let the model learn which sentences are close and which are not in relational semantics by a contrastive objective. In this process, we randomly mask entity mentions to avoid being biased by them. We show its effectiveness across several settings and benchmarks, and suggest that better pre-training technique is a reliable direction towards better RE.

2 Pilot Experiment and Analysis

To study which type of information affects existing neural RE models to make decisions, we first introduce some preliminaries of RE models and settings and then conduct pilot experiments as well as empirical analyses in this section.

2.1 Models and Dataset

There are various NRE models proposed in previous work (refer to Section 5), and we select the following three representative neural models for our pilot experiments and analyses:

CNN We use the convolutional neural networks described in Nguyen and Grishman (2015) and augment the inputs with part-of-speech, named entity recognition and position embeddings following Zhang et al. (2017).

BERT BERT is a pre-trained language model that has been widely used in NLP tasks. We use BERT for RE following Baldini Soares et al. (2019). In short, we highlight entity mentions in sentences by special markers and use the concatenations of entity representations for classification.

Matching the blanks (MTB) MTB (Baldini Soares et al., 2019) is an RE-oriented pre-trained model based on BERT. It is pre-trained by classifying whether two sentences mention the same entity pair with entity mentions randomly masked. It is fine-tuned for RE in the same way as BERT. Since it is not publicly released, we pre-train a $\text{BERT}_{\texttt{base}}$ version of MTB and give the details in Appendix A.

There are also a number of public benchmarks for RE, and we select the largest supervised RE dataset **TACRED** (Zhang et al., 2017) in our pilot experiments. TACRED is a supervised RE dataset with $106,264$ instances and 42 relations, which also provides type annotations for each entity.

Note that we use more models and datasets in our main experiments, of which we give detailed descriptions and analyses in Section 4.

2.2 Experimental Settings

We use several input formats for RE, based on which we can observe the effects of context and entity mentions in controllable experiments. The following two formats are adopted by previous literature and are close to the real-world RE scenarios:

Context+Mention (C+M) This is the most widely-used RE setting, where the whole sentence

Model	C+M	C+T	OnlyC	OnlyM	OnlyT
CNN	0.547	0.591	0.441	0.434	0.295
BERT	0.683	0.686	0.570	**0.466**	0.277
MTB	**0.691**	**0.696**	**0.581**	0.433	**0.304**

Table 1: TACRED results (micro F_1) with CNN, BERT and MTB on different settings.

(with both context and highlighted entity mentions) is provided. To let the models know where the entity mentions are, we use position embeddings (Zeng et al., 2014) for the CNN model and special entity markers (Zhang et al., 2019; Baldini Soares et al., 2019) for the pre-trained BERT.

Context+Type (C+T) We replace entity mentions with their types provided in TACRED. We use special tokens to represent them: for example, we use `[person]` and `[date]` to represent an entity with type `person` and `date` respectively. Different from Zhang et al. (2017), we do not repeat the special tokens for entity-length times to avoid leaking entity length information.

Besides the above settings, we also adopt three synthetic settings to study how much information context or mentions contribute to RE respectively:

Only Context (OnlyC) To analyze the contribution of textual context to RE, we replace all entity mentions with the special tokens `[SUBJ]` and `[OBJ]`. In this case, the information source of entity mentions is totally blocked.

Only Mention (OnlyM) In this setting, we only provide entity mentions and discard all the other textual context for the input.

Only Type (OnlyT) This is similar to **OnlyM**, except we only provide entity types in this case.

2.3 Result Analysis

Table 1 shows a detailed comparison across different input formats and models on TACRED. From the results we can see that:

(1) Both textual context and entity mentions provide critical information to support relation classification, and the most useful information in entity mentions is type information. As shown in Table 1, OnlyC, OnlyM and OnlyT suffer a significant performance drop compared to C+M and C+T, indicating that relying on only one source is not enough, and both context and entity mentions are necessary for correct prediction. Besides, we also observe that C+T achieves comparable results on TACRED with C+M for BERT and MTB. This demonstrates that most of the information provided by entity mentions is their type information. We also provide several case studies in Section 2.4, which further verify this conclusion.

C+M
Although her family was from Arkansas, *she* was born in *Washington* state, where ... **Label**: **per:state_of_birth** **Prediction**: **per:state_of_residence**
Dozens of lightly regulated subprime lenders, including New Century Financial Corp., have failed and troubled *Countrywide Financial Corp.* was acquired by *Bank of America Corp.* **Label**: **org:parents** **Prediction**: **no_relation**
C+T
First, *Natalie Hagemo* says, *she* fought the Church of Scientology just to give birth to her daughter. **Label**: **no_relation** **Prediction**: **per:children**
Earlier this week Jakarta hosted the *general assembly* of the *Organisation of Asia-Pacific News Agencies*, ... **Label**: **no_relation** **Prediction**: **org:members**
The boy, identified by the Dutch foreign ministry as *Ruben* but more fully by Dutch media as *Ruben van Assouw*, ... **Label**: **per:alternate_names** **Prediction**: **no_relation**

Table 2: Wrong predictions made only by C+M and only by C+T, where red and blue represent subject and object entities respectively. As the examples suggest, C+M is more easily biased by the entity distribution in the training set and C+T loses some information from mentions that helps to understand the text.

(2) There are superficial cues leaked by mentions in existing RE datasets, which may contribute to the high performance of RE models. We observe high performance on OnlyM with all three models on TACRED, and this phenomenon also exists in other datasets (see Table 5). We also take a deep look into the performance drop of OnlyC compared to C+M in Section 2.4, and find out that in some cases that models cannot well understand the context, they turn to rely on shallow heuristics from mentions. It inspires us to further improve models in extracting relations from context while preventing them from rote memorization of entity mentions.

We notice that CNN results are a little inconsistent with BERT and MTB: CNN on OnlyC is almost the same as OnlyM, and C+M is 5% lower than C+T. We believe that it is mainly due to the limited encoding power of CNN, which cannot fully utilize context and is more easily to overfit the shallow cues of entity mentions in the datasets.

Type	Example
Wrong 42%	..., *Jacinto Suarez*, Nicaraguan deputy to the *Central American Parliament* (PARLACEN) said Monday. **Label**: `org:top_members/employees` **Prediction**: `no_relation` US life insurance giant MetLife said on Monday it will acquire *American International Group* unit American Life Insurance company (*ALICO*) in a deal worth 155 billion dollars. **Label**: `org:subsidiaries` **Prediction**: `no_relation`
No pattern 31%	On Monday, the judge questioned the leader of the *Baptist* group, *Laura Silsby*, who ... **Label**: `per:religion` **Prediction**: `no_relation`
Confusing 27%	About a year later, *she* was transferred to Camp Hope, *Iraq*. **Label**: `per:countries_of_residence` **Prediction**: `per:stateorprovinces_of_residence`

Table 3: Case study on unique wrong predictions made by OnlyC (compared to C+M). We sample 10% of the wrong predictions, filter the wrong-labeled instances and manually annotate the wrong types to get the proportions. We use red and blue to highlight the subject and object entities.

2.4 Case Study on TACRED

To further understand how performance varies on different input formats, we carry out a thorough case study on TACRED. We choose to demonstrate the BERT examples here because BERT represents the state-of-the-art class of models and we have observed a similar result on MTB.

First we compare C+M and C+T. We find out that C+M shares 95.7% correct predictions with C+T, and 68.1% wrong predictions of C+M are the same as C+T. It indicates that most information models take advantage of from entity mentions is their type information. We also list some of the unique errors of C+M and C+T in Table 2. C+M may be biased by the entity distributions in the training set. For the two examples in Table 2, "Washington" is only involved in `per:stateorprovince_of_residence` and "Bank of America Corp." is only involved in `no_relation` in the training set, and this bias may cause the error. On the other hand, C+T may have difficulty to correctly understand the text without specific entity mentions. As shown in the example, after replacing mentions with their types, the model is confused by "general assembly" and fails to detect the relation between "Ruben" and "Ruben van Assouw". It suggests that entity mentions provide information other than types to help models understand the text.

We also study why OnlyC suffers such a significant drop compared to C+M. In Table 3, we cluster all the unique wrong predictions made by OnlyC (compared to C+M) into three classes. "Wrong" represents sentences with clear patterns but misunderstood by the model. "No pattern" means that after masking the entity mentions, it is hard to tell what relation it is even for humans. "Confusing" indicates that after masking the entities, the sentence becomes ambiguous (e.g., confusing cities and countries). As shown in Table 3, in almost half (42%) of the unique wrong predictions of OnlyC, the sentence has a clear relational pattern but the model fails to extract it, which suggests that in C+M, the model may rely on shallow heuristics from entity mentions to correctly predict the sentences. In the rest cases, entity mentions indeed provide critical information for classification.

3 Contrastive Pre-training for RE

From the observations in Section 2, we know that both context and entity type information is beneficial for RE models. However, in some cases RE models cannot well understand the relational patterns in context and rely on the shallow cues of entity mentions for classification. In order to enhance the ability to grasp entity types and extract relational facts from context, we propose the entity-masked contrastive pre-training framework for RE. We start with the motivation and process of relational contrastive example generation, and then go through the pre-training objective details.

3.1 Relational Contrastive Example Generation

We expect that by pre-training specifically towards RE, our model can be more effective at encoding relational representations from textual context and modeling entity types from mentions. To do so,

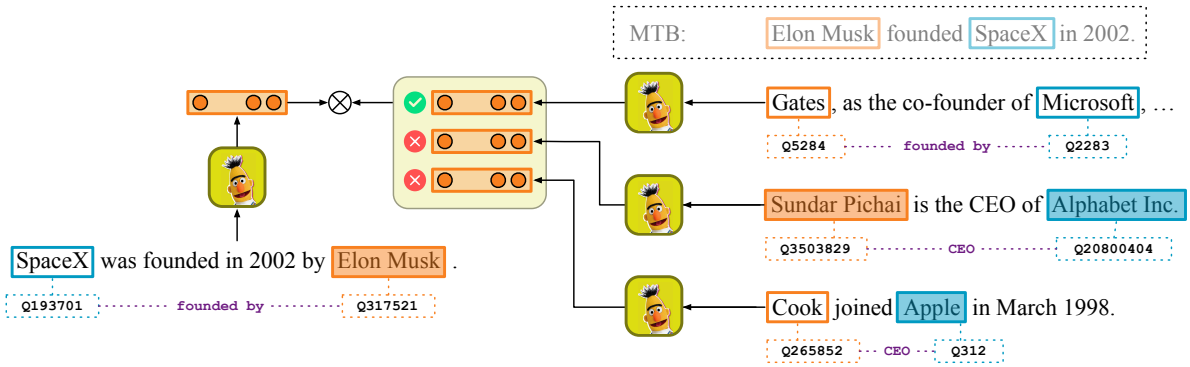

Figure 2: Our contrastive pre-training framework for RE. We assign relations to sentences by linking entity pairs in sentences to Wikidata and checking their relations in the KG. We assume that sentences with the same relation should have similar representations, and those with different relations should be pushed apart. Entity mentions are randomly masked (boxes with colored background) to avoid simple memorization. Compared to MTB (in the dotted box), our method samples data with better diversity, which can not only increase the coverage of entity types and diverse context but also reduce the possibility of memorizing entity names.

we adopt the idea of contrastive learning (Hadsell et al., 2006), which aims to learn representations by pulling "neighbors" together and pushing "non-neighbors" apart. After this, "neighbor" instances will have similar representations. So it is important to define "neighbors" in contrastive learning and we utilize the information from KGs to to that. Inspired by distant supervision (Mintz et al., 2009), we assume that sentences with entity pairs sharing the same relation in KGs are "neighbors".

Formally, denote the KG we use as $\mathcal{K}$, which is composed of relational facts. Denote two random sentences as X_A and X_B, which have entity mentions h_A, t_A and h_B, t_B respectively. We define X_A and X_B as "neighbors" if there is a relation r such that $(h_A, r, t_A) \in \mathcal{K}$ and $(h_B, r, t_B) \in \mathcal{K}$. We take Wikidata as the KG since it can be easily linked to the Wikipedia corpus used for pre-training. When training, we first sample a relation r with respect to its proportion in the KG, and then sample a sentence pair (X_A, X_B) linked to r. To learn contrastively, we randomly sample N sentences $X_B^i, 1 \leq i \leq N$ so they can form N negative pairs with X_A. The model needs to classify which sentence among all the postive and negative samples has the same relation with X_A.

To avoid memorizing entity mentions or extracting shallow features from them during pre-training, we randomly mask entity mentions with the special token `[BLANK]`. We use $P_{\texttt{BLANK}}$ to denote the ratio of replaced entities and set $P_{\texttt{BLANK}} = 0.7$ following Baldini Soares et al. (2019). Note that masking all mentions during pre-training is also not a good option since it will create a gap between pre-training and fine-tuning and also block the pre-trained models from utilizing entity mention information (e.g., learning entity types).

Take an example to understand our data generation process: In Figure 2, there are two sentences "*SpaceX was founded in 2002 by Elon Musk*" and "*As the co-founder of Microsoft, Bill Gates ...*" where both (SpaceX, `founded by`, Elon Musk) and (Microsoft, `founded by`, Bill Gates) exist in the KG. We expect the two sentences to have similar representations reflecting the relation. On the other hand, for the other two sentences in the right part of the figure, since their entity pairs do not have the relation `founded by`, they are regarded as negative samples and are expected to have diverse representations from the left one. During pre-training, each entity mention has a probability of $P_{\texttt{BLANK}}$ to be masked.

The main problem of the generation process is that the sentence may express no relation between the entities at all, or express the relation different from what we expect. For example, a sentence mentioning "SpaceX" and "Elon Musk" may express the relation `founded by`, `CEO` or `CTO`, or simply does not express any relation between them. An example could be "*Elon Musk answers reporters' questions on a SpaceX press conference*", which expresses no clear relation between the two. However, we argue that the noise problem is not critical for our pre-training framework: Our goal is to get relatively better representations towards RE compared to raw pre-trained models like BERT, rather than to directly train an RE model for downstream tasks, so noise in the data is acceptable.

Dataset	# Rel.	# Inst.	% N/A
TACRED	42	106,264	79.5%
SemEval-2010 Task 8	19	10,717	17.4%
Wiki80	80	56,000	-
ChemProt	13	10,065	-
FewRel	100	70,000	-

Table 4: Statistics for RE datasets used in the paper, including numbers of relations, numbers of instances and proportions of `N/A` instances. "-" for the last column means that there is no `N/A` relation in the dataset.

With the help of the generated relational contrastive examples, our model can learn to better grasp type information from mentions and extract relational semantics from textual context: (1) The paired two sentences, which mention different entity pairs but share the same relation, prompt the model to discover the connections between these entity mentions for the relation. Besides, the entity masking strategy can effectively avoid simply memorizing entities. This eventually encourages the model to exploit entity type information. (2) Our generation strategy provides a diverse set of textual context expressing the same relation to the model, which motivates the model to learn to extract the relational patterns from a variety of expressions.

Compared with our model, MTB (Baldini Soares et al., 2019) takes a more strict rule which requires the two sampled sentences to share the same entity pair. While it reduces the noise, the model also samples data with less diversity and loses the chance to learn type information.

3.2 Training Objectives

In our contrastive pre-training, we use the same Transformer architecture (Vaswani et al., 2017) as BERT. Denote the Transformer encoder as `ENC` and the output at the position i as $\texttt{ENC}_i(\cdot)$. For the input format, we use special markers to highlight the entity mentions following Baldini Soares et al. (2019). For example, for the sentence "*SpaceX was founded by Elon Musk*.", the input sequence is "`[CLS] [E1]` SpaceX `[/E1]` was founded by `[E2]` Elon Musk `[/E2] . [SEP]`".

During the pre-training, we have two objectives: contrastive pre-training objective and masked language modeling objective.

Contrastive Pre-training Objective As shown in Figure 2, given the positive sentence pair (x_A, x_B), and negative sentence pairs $(x_A, x_B^i), 1 \leq i \leq N$, we first use the Transformer encoder to get relation-aware representation for x in $\{x_A, x_B\} \cup \{x_B^i\}_{i=1}^N$:

$$\mathbf{x} = \texttt{ENC}_h(x) \oplus \texttt{ENC}_t(x), \tag{1}$$

where h and t are the positions of special tokens `[E1]` and `[E2]`, and $\oplus$ stands for concatenation. With the sentence representation, we have the following training objective:

$$\mathcal{L}_{CP} = -\log \frac{e^{\mathbf{x}_A^T \mathbf{x}_B}}{e^{\mathbf{x}_A^T \mathbf{x}_B} + \sum_{i=1}^{i \leq N} e^{\mathbf{x}_A^T \mathbf{x}_B^i}}. \tag{2}$$

By optimizing the model with respect to $\mathcal{L}_{CP}$, we expect representations for x_A and x_B to be closer and eventually sentences with similar relations will have similar representations.

Masked Language Modeling Objective To maintain the ability of language understanding inherited from BERT and avoid catastrophic forgetting (McCloskey and Cohen, 1989), we also adopt the masked language modeling (MLM) objective from BERT. MLM randomly masks tokens in the inputs and by letting the model predict the masked tokens, MLM learns contextual representation that contains rich semantic and syntactic knowledge. Denote the MLM loss as $\mathcal{L}_{MLM}$.

Eventually, we have the following training loss:

$$\mathcal{L} = \mathcal{L}_{CP} + \mathcal{L}_{MLM}. \tag{3}$$

4 Experiment

In this section, we explore the effectiveness of our relational contrastive pre-training across two typical RE tasks and several RE datasets.

4.1 RE Tasks

For comprehensive experiments, we evaluate our models on various RE tasks and datasets.

Supervised RE This is the most widely-adopted setting in RE, where there is a pre-defined relation set $\mathcal{R}$ and each sentence x in the dataset expresses one of the relations in $\mathcal{R}$. In some benchmarks, there is a special relation named `N/A` or `no_relation`, indicating that the sentence does not express any relation between the given entities, or their relation is not included in $\mathcal{R}$.

For supervised RE datasets, we use TACRED (Zhang et al., 2017), SemEval-2010 Task 8 (Hendrickx et al., 2009), Wiki80 (Han et al., 2019) and ChemProt (Kringelum et al., 2016). Table 4 shows the comparison between the datasets.

Dataset	Model	1%			10%			100%		
		C+M	OnlyC	OnlyM	C+M	OnlyC	OnlyM	C+M	OnlyC	OnlyM
TACRED	BERT	0.211	0.167	0.220	0.579	0.446	0.433	0.683	0.570	**0.466**
	MTB	0.304	0.231	0.308	0.608	0.496	0.441	0.691	0.581	0.433
	CP	**0.485**	**0.393**	**0.350**	**0.633**	**0.515**	**0.453**	**0.695**	**0.593**	0.450
SemEval	BERT	0.367	0.294	0.245	0.772	0.688	0.527	0.871	0.798	0.677
	MTB	0.362	0.330	**0.249**	0.806	0.744	0.543	0.873	0.807	**0.682**
	CP	**0.482**	**0.470**	0.221	**0.822**	**0.766**	**0.543**	**0.876**	**0.811**	0.679
Wiki80	BERT	0.559	0.413	0.463	0.829	0.413	0.655	0.913	0.810	0.781
	MTB	0.585	0.509	0.542	0.859	0.509	0.719	0.916	0.820	0.788
	CP	**0.827**	**0.734**	**0.653**	**0.893**	**0.734**	**0.745**	**0.922**	**0.834**	**0.799**
ChemProt	BERT	0.362	0.362	0.362	0.634	0.584	0.385	0.792	0.777	0.463
	MTB	**0.362**	0.362	**0.362**	0.682	0.685	0.403	0.796	0.798	0.463
	CP	0.361	**0.362**	0.360	**0.708**	**0.697**	**0.404**	**0.806**	**0.803**	**0.467**

Table 5: Results on supervised RE datasets TACRED (micro F_1), SemEval (micro F_1), Wiki80 (accuracy) and ChemProt (micro F_1). 1% / 10% indicate using 1% / 10% supervised training data respectively.

We also add 1% and 10% settings, meaning using only 1% / 10% data of the training sets. It is to simulate a low-resource scenario and observe how model performance changes across different datasets and settings. Note that ChemProt only has $4,169$ training instances, which leads to the abnormal results on 1% ChemProt in Table 5. We give details about this problem in Appendix B.

Few-Shot RE Few-shot learning is a recently emerged topic to study how to train a model with only a handful of examples for new tasks. A typical setting for few-shot RE is N-way K-shot RE (Han et al., 2018), where for each evaluation episode, N relation types, K examples for each type and several query examples (all belonging to one of the N relations) are sampled, and models are required to classify the queries based on given $N \times K$ samples. We take FewRel (Han et al., 2018; Gao et al., 2019) as the dataset and list its statistics in Table 4.

We use Prototypical Networks as in Snell et al. (2017); Han et al. (2018) and make a little change: (1) We take the representation as described in Section 3.2 instead of using `[CLS]`. (2) We use dot production instead of Euclidean distance to measure the similarities between instances. We find out that this method outperforms original Prototypical Networks in Han et al. (2018) by a large margin.

4.2 RE Models

Besides BERT and MTB we have introduced in Section 2.1, we also evaluate our proposed **c**ontrastive **p**re-training framework for RE (CP). We write the detailed hyper-parameter settings of both the pre-training and fine-tuning process for all the models in Appendix A and B.

Note that since MTB and CP use Wikidata for pre-training, and Wiki80 and FewRel are constructed based on Wikidata, we exclude all entity pairs in test sets of Wiki80 and FewRel from pre-training data to avoid test set leakage.

4.3 Strength of Contrastive Pre-training

Table 5 and 6 show a detailed comparison between BERT, MTB and our proposed contrastive pre-trained models. Both MTB and CP improve model performance across various settings and datasets, demonstrating the power of RE-oriented pre-training. Compared to MTB, CP has achieved even higher results, proving the effectiveness of our proposed contrastive pre-training framework. To be more specific, we observe that:

(1) CP improves model performance on all C+M, OnlyC and OnlyM settings, indicating that our pre-training framework enhances models on both context understanding and type information extraction.

(2) The performance gain on C+M and OnlyC is universal, even for ChemProt and FewRel 2.0, which are from biomedical domain. Our models trained on Wikipedia perform well on biomedical datasets, suggesting that CP learns relational patterns that are effective across different domains.

(3) CP also shows a prominent improvement of OnlyM on TACRED, Wiki80 and FewRel 1.0, which are closely related to Wikipedia. It indicates that our model has a better ability to extract type information from mentions. Both promotions on context and mentions eventually lead to better RE results of CP (better C+M results).

(4) The performance gain made by our contrastive pre-training model is more significant on

Model	5-way 1-shot			5-way 5-shot			10-way 1-shot			10-way 5-shot		
	C+M	OnlyC	OnlyM	C+M	OnlyC	OnlyM	C+M	OnlyC	OnlyM	C+M	OnlyC	OnlyM
FewRel 1.0												
BERT	0.911	0.866	0.701	0.946	0.925	0.804	0.842	0.779	0.575	0.908	0.876	0.715
MTB	0.911	0.879	0.727	0.954	0.939	0.835	0.843	0.779	0.568	0.918	0.892	0.742
CP	**0.951**	**0.926**	**0.743**	**0.971**	**0.956**	**0.840**	**0.912**	**0.867**	**0.620**	**0.947**	**0.924**	**0.763**
FewRel 2.0 Domain Adaptation												
BERT	0.746	0.683	0.316	0.827	0.782	0.406	0.635	0.542	0.210	0.765	0.706	0.292
MTB	0.747	0.692	**0.338**	**0.879**	0.836	0.426	0.625	0.528	**0.216**	**0.811**	**0.744**	**0.298**
CP	**0.797**	**0.745**	0.335	0.849	**0.840**	**0.437**	**0.681**	**0.601**	0.213	0.798	0.738	0.297

Table 6: Accuracy on FewRel dataset. FewRel 1.0 is trained and tested on Wikipedia domain. FewRel 2.0 is trained on Wikipedia domain but tested on biomedical domain.

low-resource and few-shot settings. For C+M, we observe a promotion of 7% on 10-way 1-shot FewRel 1.0, 18% improvement on 1% setting of TACRED, and 24% improvement on 1% setting of Wiki80. There is also a similar trend for OnlyC and OnlyM. In the low resource and few-shot settings, it is harder for models to learn to extract relational patterns from context and easier to overfit to superficial cues of mentions, due to the limited training data. However, with the contrastive pre-training, our model can relatively take better use of textual context while avoiding being biased by entities, and outperform the other baselines by a large margin.

5 Related Work

Development of RE RE of early days has gone through pattern-based methods (Huffman, 1995; Califf and Mooney, 1997), feature-based methods (Kambhatla, 2004; Zhou et al., 2005), kernel-based methods (Culotta and Sorensen, 2004; Bunescu and Mooney, 2005), graphical models (Roth and Yih, 2002, 2004), etc. Since Socher et al. (2012) propose to use recursive neural networks for RE, there have been extensive studies on neural RE (Liu et al., 2013; Zeng et al., 2014; Zhang and Wang, 2015). To solve the data deficiency problem, researchers have developed two paths: **distant supervision** (Mintz et al., 2009; Min et al., 2013; Riedel et al., 2010; Zeng et al., 2015; Lin et al., 2016) to automatically collect data by aligning KGs and text, and **few-shot learning** (Han et al., 2018; Gao et al., 2019) to learn to extract new relations by only a handful of samples.

Pre-training for RE With the recent advance of pre-trained language models (Devlin et al., 2019), applying BERT-like models as the backbone of RE systems (Baldini Soares et al., 2019) has become a standard procedure. Based on BERT, Baldini Soares et al. (2019) propose matching the blanks, an RE-oriented pre-trained model to learn relational patterns from text. A different direction is to inject entity knowledge, in the form of entity embeddings, into BERT (Zhang et al., 2019; Peters et al., 2019; Liu et al., 2020). We do not discuss this line of work here for their promotion comes from relational knowledge of external sources, while we focus on text itself in the paper.

Analysis of RE Han et al. (2020) suggest to study how RE models learn from context and mentions. Alt et al. (2020) also point out that there may exist shallow cues in entity mentions. However, there have not been systematical analyses about the topic and to the best of our knowledge, we are the first one to thoroughly carry out these studies.

6 Conclusion

In this paper, we thoroughly study how textual context and entity mentions affect RE models respectively. Experiments and case studies prove that (i) both context and entity mentions (mainly as type information) provide critical information for relation extraction, and (ii) existing RE datasets may leak superficial cues through entity mentions and models may not have the strong abilities to understand context as we expect. From these points, we propose an entity-masked contrastive pre-training framework for RE to better understand textual context and entity types, and experimental results prove the effectiveness of our method.

In the future, we will continue to explore better RE pre-training techniques, especially with a focus on open relation extraction and relation discovery. These problems require models to encode good relational representation with limited or even zero annotations, and we believe that our pre-trained RE models will make a good impact in the area.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2018YFB1004503), the National Natural Science Foundation of China (NSFC No. 61532010) and Beijing Academy of Artificial Intelligence (BAAI). This work is also supported by the Pattern Recognition Center, WeChat AI, Tencent Inc. Gao is supported by 2019 Tencent Rhino-Bird Elite Training Program. Gao is also supported by Tsinghua University Initiative Scientific Research Program.

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of ACL*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of ACL*, pages 2895–2905.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of EMNLP*, pages 615–620.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of EMNLP*, pages 724–731.

Mary Elaine Califf and Raymond J. Mooney. 1997. Relational learning of pattern-match rules for information extraction. In *Proceedings of CoNLL*.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of ACL*, page 423.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of EMNLP-IJCNLP*, pages 6251–6256.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2020. More data, more relations, more context and more openness: A review and outlook for relation extraction.

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. OpenNRE: An open and extensible toolkit for neural relation extraction. In *Proceedings of EMNLP-IJCNLP: System Demonstrations*, pages 169–174.

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 EMNLP*, pages 4803–4809.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 94–99.

Scott B Huffman. 1995. Learning information extraction patterns from examples. In *Proceedings of IJCAI*, pages 246–260.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of ACL*, pages 178–181.

Jens Kringelum, Sonny Kim Kjaerulff, Søren Brunak, Ole Lund, Tudor I Oprea, and Olivier Taboureau. 2016. ChemProt-3.0: A global chemical biology diseases mapping. *Database*, 2016.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*, pages 2124–2133.

Chunyang Liu, Wenbo Sun, Wenhan Chao, and Wanxiang Che. 2013. Convolution neural network for relation extraction. In *Proceedings of ICDM*, pages 231–242.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling language representation with knowledge graph. In *Proceedings of AAAI*, pages 2901–2908.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR 2019*.

Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Mem2Seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *Proceedings of ACL*, pages 1468–1478.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of NAACL*, pages 777–782.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*, pages 1003–1011.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the NAACL Workshop on Vector Space Modeling for NLP*, pages 39–48.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of EMNLP-IJCNLP*, pages 43–54.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of EMNLP-IJCNLP*, pages 2463–2473.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML-PKDD*, pages 148–163.

Dan Roth and Wen-tau Yih. 2002. Probabilistic reasoning for entity & relation recognition. In *Proceedings of COLING*.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of CoNLL*.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of NIPS*, pages 4077–4087.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP*, pages 1201–1211.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Proceedings of CACM*, 57(10):78–85.

Chenyan Xiong, Russell Power, and Jamie Callan. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In *Proceedings of WWW*, pages 1271–1279.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*, pages 1753–1762.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*, pages 2335–2344.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of EMNLP*, pages 35–45.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of ACL*, pages 1441–1451.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of ACL*, pages 427–434.

A Pre-training Details

Pre-training Dataset We construct a dataset for pre-training following the method in the paper. We use Wikipedia articles as corpus and Wikidata (Vrandečić and Krötzsch, 2014) as the knowledge graph. Firstly, We use anchors to link entity mentions in Wikipedia corpus with entities in Wikidata. Then, in order to link more unanchored entity mentions, we adopt `spaCy`[1] to find all possible entity mentions, and link them to entities in Wikidata via name matching. Finally, we get a pre-training dataset containing 744 relations and 867,278 sentences. We release this dataset together with our source code at our GitHub repository[2].

We also use this dataset for MTB, which is slightly different from the original paper (Baldini Soares et al., 2019). The original MTB takes all entity pairs into consideration, even if they do not have a relationship in Wikidata. Using the above dataset means that we filter out these entity pairs. We do this out of training efficiency, for those entity pairs that do not have a relation are likely to express little relational information, and thus contribute little to the pre-training.

Data Sampling Strategy For MTB (Baldini Soares et al., 2019), we follow the same sampling strategy as in the original paper. For pre-training our contrastive model, we regard sentences labeled with the same relation as a "bag". Any sentence pair whose sentences are in the same bag is treated as a positive pair and as a negative pair otherwise. So there will be a large amount of possible positive samples and negative samples. We dynamically sample positive pairs of a relation with respect to the number of sentences in the bag.

Hyperparameters We use Huggingface's `Transformers`[3] to implement models for both pre-training and fine-tuning and use AdamW (Loshchilov and Hutter, 2019) for optimization. For most pre-training hyperparameters, we select the same values as Baldini Soares et al. (2019). We search hyperparameter batch size in $\{256, 2048\}$ and $P_{\texttt{BLANK}}$ in $\{0.3, 0.7\}$. For MTB, batch size N means that a batch contains $2N$ sentences, which form $N/2$ positive pairs and $N/2$ negative pairs. For CP, batch size N means that a batch contains $2N$ sentences, which form N positive pairs. For negative samples, we pair the sentence in each pair with sentences in other pairs.

Parameter	MTB	CP
Learning Rate	3×10^{-5}	3×10^{-5}
Batch Size	256	2048
Sentence Length	64	64
$P_{\texttt{BLANK}}$	0.7	0.7

Table 7: Hyperparameters for pre-training models. $P_{\texttt{BLANK}}$ corresponds to the probability of replacing entities with `[BLANK]`.

Dataset	Train	Dev	Test
TACRED	68,124	22,631	15,509
SemEval	6,507	1,493	2,717
Wiki80	39,200	5,600	11,200
ChemProt	4,169	2,427	3,469
FewRel	44,800	11,200	14,000

Table 8: Numbers of instances in train / dev / test splits for different RE datasets.

We set hyperparameters according to results on supervised RE dataset TACRED (micro F_1). Table 7 shows hyperparameters for pre-training MTB and our contrastive model (CP). The batch size of our implemented MTB is different from that in Baldini Soares et al. (2019), because in our experiments, MTB with a batch size of 256 performs better on TACRED than the batch size of 2048.

Pre-training Efficiency MTB and our contrastive model have the same architecture as $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019), so they both hold 110M parameters approximately. We use four Nvidia 2080Ti GPUs to pre-train models. Pre-training MTB takes 30,000 training steps and approximately 24 hours. Pre-training our model takes 3,500 training steps and approximately 12 hours.

B RE Fine-tuning

RE Datasets We download TACRED from `LDC`[4], Wiki80, SemEval from `OpenNRE`[5], ChemProt from `sciBert`[6], and FewRel from `FewRel`[7]. Table 8 shows detailed statistics for each dataset and Table 9 demonstrates the sizes of training data for different supervised RE datasets

[1] https://spacy.io/

[2] https://github.com/thunlp/RE-Context-or-Names

[3] https://github.com/huggingface/transformers

[4] https://catalog.ldc.upenn.edu/LDC2018T24

[5] https://github.com/thunlp/OpenNRE

[6] https://github.com/allenai/scibert

[7] https://github.com/thunlp/fewrel

Dataset	1%	10%	100%
TACRED	703	6,833	68,124
SemEval	73	660	6,507
Wiki80	400	3,920	3,9200
ChemProt	49	423	4,169

Table 9: Numbers of training instances in supervised RE datasets under different proportion settings.

Parameter	Supervised RE	Few-Shot RE
Learning Rate	3×10^{-5}	2×10^{-5}
Batch Size	64	4
Epoch	6	10
Sentence Length	100	128
Hidden Size	768	768

Table 10: Hyperparameters for fine-tuning on relation extraction tasks (BERT, MTB and CP).

in 1%, 10% and 100% settings. For 1% and 10% settings, we randomly sample 1% and 10% training data for each relation (so the total training instances for 1% / 10% settings are not exactly 1% / 10% of the total training instances in the original datasets). As shown in the table, the numbers of training instances in SemEval and ChemProt for 1% setting are extremely small, which explains the abnormal performance.

Hyperparameters Table 10 shows hyperparameters when finetuning on different RE tasks for BERT, MTB and CP. For CNN, we train the model by SGD with a learning rate of 0.5, a batch size of 160 and a hidden size of 230. For few-shot RE, we use the recommended hyperparameters in `FewRel`[8].

Multiple Trial Settings For all the results on supervised RE, we run each experiment 5 times using 5 different seeds $(42, 43, 44, 45, 46)$ and select the median of 5 results as the final reported number. For few-shot RE, as the model varies little with different seeds and it is evaluated in a sampling manner, we just run one trial with 10000 evaluation episodes, which is large enough for the result to converge. We report accuracy (proportion of correct instances in all instances) for Wiki80 and FewRel, and micro F_1[9] for all the other datasets.

[8] https://github.com/thunlp/FewRel

[9] https://en.wikipedia.org/wiki/F1_score