Rethinking the Role of Entity Type in Relation Classification

Xiang Dai **Sarvnaz Karimi** **Stephen Wan**
CSIRO Data61
{dai.dai,sarvnaz.karimi,stephen.wan}@csiro.au

Abstract

Relation Classification (RC)—the task of identifying the relation between a pair of target entities—is a fundamental sub-task of information extraction. RC models built on top of entity information are prevalent, with different variants using entity information, especially entity type information, differently. However, RC models are often benchmarked on datasets that human annotators provide near-perfect entity information, and, state-of-the-art results are reported using gold entity type information. We believe there is a need to understand how the effectiveness of RC models is affected by the correctness of entity type information because in practice this information is provided by imperfect entity recognition models. Our results on six datasets across four domains show that although using gold entity type improves the effectiveness of RC models, incorrect entity types may cause large effectiveness drops on some (but not all) datasets. We propose using Pointwise Mutual Information (PMI) to identify datasets on which RC models may be negatively impacted by incorrect entity type information.

1 Introduction

Relation extraction is a fundamental sub-task of information extraction that aims to extract structured information from unstructured text. It can be useful for many downstream applications, such as opinion mining, question answering, and knowledge graph construction (Choi et al., 2006; Ji and Grishman, 2011; Nickel et al., 2015; Zhang et al., 2019a). One common approach to relation extraction is pipeline-based, where Named Entity Recognition (NER) models are first used to identify entity names in text and then the identified entities are fed into a Relation Classification (RC) model, identifying the relation between a pair of target entities (See an example in Figure 1).

Previous studies (Soares et al., 2019; Peng et al., 2020; Zhong and Chen, 2021; Zhou and Chen, 2022) show that adding information on entity *position* and *type* is critical for the RC models to learn useful relation representations, and the RC model heavily relies on the entity information, especially entity type information. However, the effectiveness of proposed methods of incorporating entity information into the RC model is largely benchmarked on datasets where human-annotated entity information is provided. Start-of-the-art results are reported with RC models using gold entity type information (Lyu and Chen, 2021; Zhou and Chen, 2022; Han et al., 2022). There is a gap in the literature to investigate how the effectiveness of RC models is affected by the correctness of entity type information. In other words, given that no NER model is perfect, how may the availability of accurate entity type information affect our choice of RC models?

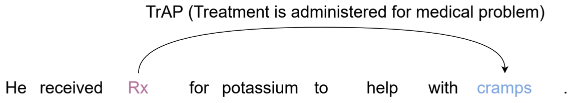

TrAP (Treatment is administered for medical problem)
He received Rx for potassium to help with cramps .

Figure 1: An example taken from I2B2-2010 (Uzuner et al., 2011). It contains the 'TrAP' relation between two entities: Rx (entity type: Treatment) and cramps (entity type: Problem). Note that this sentence contains another relation between 'potassium' and 'cramps'.

To answer this research question, we present the following contributions:

- We compare different approaches of Transformer-based RC models that incorporate entity type information via minimal architecture change. Based on experimental results on six datasets across four domains, we find that incorporating *gold* entity type information using special markers outperforms other approaches using entity type embeddings or entity type as part of the initial decoder input.
- We conduct a sensitivity analysis of the RC

Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pages 374–384
November 1–4, 2023. ©2023 Association for Computational Linguistics

model with respect to the correctness of entity type information. Our results show that entity type errors may cause a large effectiveness drop on some (but not all) datasets, and this phenomenon may change the decision of how to incorporate entity type information to RC models.

- We show that Pointwise Mutual Information (PMI) can be used to identify datasets on which RC models may be negatively impacted by incorrect entity types and help decide how to use entity type for the RC model.

2 Related work

In earlier literature, Relation Classification (RC) models rely on manually defined features (Craven and Kumlien, 1999; Mintz et al., 2009), convolutional neural network (Zeng et al., 2014; dos Santos et al., 2015), recurrent neural network (Zhang and Wang, 2015; Miwa and Bansal, 2016) or graph neural network (Guo et al., 2019) to build relation representation. To effectively capture the interaction between entities, in addition to entity information, these models either explicitly make use of syntactic information (Mintz et al., 2009) or use neural networks to learn context information (Vu et al., 2016; Sorokin and Gurevych, 2017).

After the introduction of BERT (Devlin et al., 2019), the pre-training-then-fine-tuning paradigm dominated. RC models based on pre-trained language representation models have also gained significant success (Wu and He, 2019; Alt et al., 2019; Wei et al., 2019). Recent research can be divided into three groups. One research direction continues to improve pre-trained models via injecting factual and linguistic knowledge, usually with the help of external knowledge base consisting of relation tuples (Peters et al., 2019; Soares et al., 2019; Zhang et al., 2019b; Yamada et al., 2020; Wang et al., 2021). Another line of research designs specialised pre-training objectives to help better modelling spans, which usually refer to entities (Joshi et al., 2020; Lin et al., 2021). The last category focuses on the fine-tuning stage where modifications are proposed to incorporate syntactic features (Adel and Strötgen, 2021) and entity information (Bilan and Roth, 2018; Eberts and Ulges, 2020; Li et al., 2020; Zhou and Chen, 2022; Han et al., 2022).

Our study falls in the last category, and we focus on analysing how the correctness of entity types may impact the effectiveness of RC models.

Although we focus on a pipeline-based approach for relation extraction, our work also relates to another group of relation extraction methods that model NER and RC jointly (Miwa and Bansal, 2016; Lin et al., 2020; Eberts and Ulges, 2020; Yan et al., 2022). On the one hand, both approaches employ methods of incorporating entity information into RC, and design options can be shared. On the other hand, although joint models aim to mitigate error propagation via modelling entity and relation representations together, they still rely on ground truth entity information for training the relation component. For example, Eberts and Ulges (2020) train the relation classifier via drawing negative examples from gold entity pairs that are not labelled with any relation. We believe our analysis can provide insights into designing components used in joint models when high-quality entity information is unavailable.

Debates about the usefulness of entity type information for RC models Peng et al. (2020) observe that Context+EntityType—replacing entity names with their entity types—achieves comparable results on TACRED with Context+EntityName for BERT. They argue that *using original entity names may be biased by the entity distributions in the training set* and the RC models may not generalise well to unseen entities. By the same consideration, Zhang et al. (2018); Joshi et al. (2020) use entity types to replace entity names. Zhou and Chen (2022) argue that if the RC models should not consider entity names, it is unreasonable to suppose that they can be improved by external knowledge graphs, which is an active research area in the literature. They propose to insert special typed markers around original entity names (detailed in Section 3.2) and show that the proposed variant achieves state-of-the-art effectiveness on multiple datasets. However, Zhou and Chen notice that using entity type information brings smaller improvements on a clean test set than a noisy test set. They hypothesise that this result may be attributed to *annotation biases*. That is, some annotators may label the relation only based on target entities without reading the context. The paradigm proposed by Lyu and Chen (2021) makes stronger assumptions about the correctness of entity types, which are used to *filter candidate relations*. A specific classifier is individually learned for each pair of entity types to predict a specific set of candidate relations.

3 Preliminaries of RC Models

Problem Formulation Relation classification is framed as a task where given a text sequence $\mathcal{X} = [x_0, \cdots, x_n]$ and two entity names e_1 and e_2, the RC model predicts either (1) a relation $r \in \mathcal{R}$ that holds between two entities; or, (2) NA relation (no relation or none of the pre-defined relation hold). We aim to investigate how to effectively incorporate entity types of e_1 and e_2 in RC models, and how the effectiveness of the RC model is affected by the correctness of these entity types.

In the following, we first group existing RC models into three categories: span-based, marker-based and prompt-based, and then describe how entity type can be incorporated into these models.

3.1 Span-based Models

The span-based RC models usually consist of three components: (1) a token encoder, (2) a relation encoder, and (3) a classifier. The token encoder takes $\mathcal{X}$ as input and generates a list of contextual token representations $\mathcal{H} = [h_0, \cdots, h_n]$.

After the contextual token presentations $\mathcal{H}$ are obtained from the token encoder, span-based models first build span (e.g., entity names, the context span between two target entities) representations from $\mathcal{H}$. There are many options for the fusion function proposed in the literature. For example, Eberts and Ulges (2020) max-pool contextual token representations to obtain span embeddings; Wu and He (2019) apply the average operation to obtain span embeddings; Yu et al. (2020) use biaffine attention to build span embeddings; and, concatenating token representations corresponding to boundary tokens for span embeddings (Joshi et al., 2020). Our preliminary experiments find that max-pooling (Eberts and Ulges, 2020) performs best, although the difference between these variants is very small.

Eberts and Ulges (2020) propose to concatenate three span embeddings—corresponding to two entities and the context between them—as the relation representation $\hbar$. We also investigate concatenating more spans (e.g., context before the first entity names and context after the last entity names) or the hidden states corresponding to the [CLS] token, but find that these variants do not improve the effectiveness of the RC models. We denote the model variant by Eberts and Ulges (2020) as **SpU** in experimental results (Table 3).

Incorporating entity type information via segment embeddings To provide the RC model with entity type information, we propose incorporating segment embedding into the input of the token encoder (Sorokin and Gurevych, 2017). We first mark each token in the entity names using their entity types. An embedding matrix, $\boldsymbol{E} \in \mathbb{R}^{(c+1)\times 768}$ (c is the total number of entity types), is then used to convert these entity types into the segment embedding, and finally, we sum the segment, token, and position embeddings and feed them into the token encoder. In our preliminary experiments, we also investigate concatenating segment embeddings with token encoder outputs but find it underperforms the variant where segment embeddings are fed into the token encoder. We denote this variant as **SpT** in Table 3.

3.2 Marker-based Models

Methods belonging to this category usually modify the original text sequence by either inserting special markers or using special markers to replace original entity names. Then the hidden states corresponding to these special markers are used to build the relation representation. For example, Soares et al. (2019) insert special markers, i.e., [E1], [/E1], [E2] and [/E2], before and after target entities, and then concatenate hidden states corresponding to [E1] and [E2] as the relation representation. Arguing that these newly introduced markers, such as [E1] and [E2], are not well pre-trained, Zhou and Chen (2022) propose to use punctuation markers such as @ and # to enclose target entities. Zhou and Chen also use special markers to incorporate the entity type. That is, they use $*$ and $\wedge$ to enclose entity type and prepend to entity names. We denote the variant of using untyped markers as **MaU** and

MaU	received @ Rx @ for potassium to help with # cramps #
MaTi	received @ * treatment * Rx @ for potassium to help with # ∧ problem ∧ cramps #
MaTr	received @ * treatment * @ for potassium to help with # ∧ problem ∧ #

Table 1: Examples of the modified input. **MaU**: untyped marker; **MaTi**: typed marker (insert); **MaTr**: typed marker (replace). The hidden states of underlined tokens are concatenated and used as the relation representation.

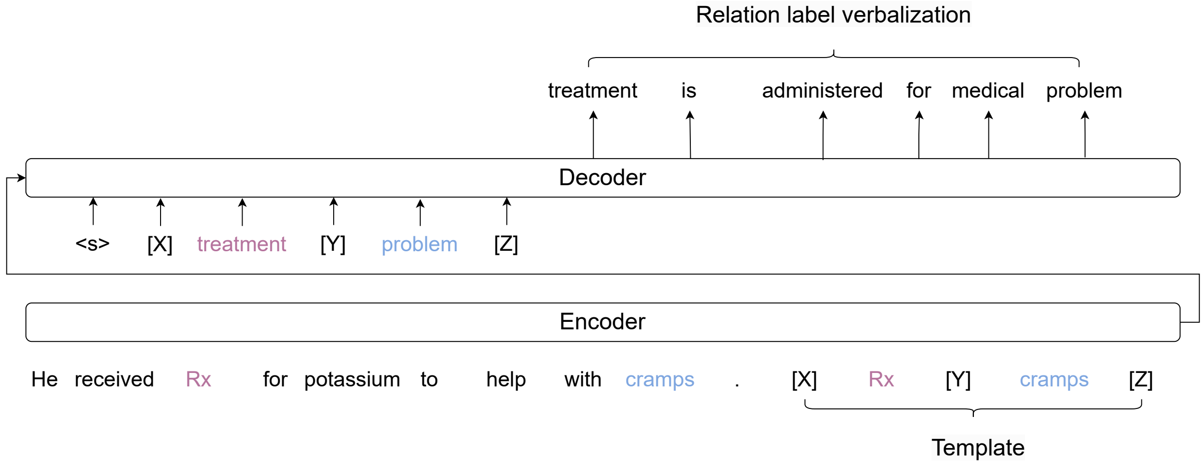


Figure 2: A high-level illustration of prompt-based RC model. Note that [X], [Y], [Z] are three sentinel tokens in the template, and we omit trainable prompt embeddings in the template for the sake of simplicity. We refer readers to (Han et al., 2022) for more details.

the one using typed markers as **MaTi** in Table 3, and examples of the modified text are shown in Table 1.

Zhang et al. (2017); Joshi et al. (2020) replace entities by their entity types such as '[SUBJ-TYPE]' and '[OBJ-TYPE]' and predict the relation type using the hidden states of [CLS] token. We find concatenating hidden states corresponding to two beginning markers—punctuation markers instead of newly introduced markers—performs better, and we denote this variant as **MaTr** (See example in Table 1).

3.3 Prompt-based Models

Prompt-based models employ encoder-decoder architecture and convert the classification problem to a text generation problem (Han et al., 2022; Chen et al., 2022; Xu et al., 2022). That is, the original text sequence $\mathcal{X}$ is reformulated by adding a cloze-style phrase called *template*. The modified text sequence is then taken as the input of the seq2seq model, and the model generates a sequence of tokens called *relation label verbalisation* and can be mapped from relation labels $\mathcal{R}$. See Figure 2 for a high-level illustration.

Instead of the handcrafted template (such as 'The relation between Rx and cramps is <mask>', Han et al. (2022) design a method that uses a series of learnable continuous tokens as prompts. They copy target entity names after the original text sequence and use three sentinel tokens ([X], [Y], [Z]) to separate target entity names in the template (See example in Figure 2). Then, the original text sequence and the template are mapped to a sequence of continuous vectors via the token embedding layer. After this transformation, a few learnable vectors, which are jointly optimised by gradient descent, are inserted in front of token embeddings corresponding to these three sentinel tokens. Finally, the new sequence of token embeddings, which is summed together with the position embedding, is fed into the encoder.

To use entity type information to influence the choice of possible candidate relations, Han et al. append the entity type tokens as part of the initial decoder inputs. We denote this variant called GenPT as **PrT** and the variant without entity type information—the initial decoder input is '<s> [Z]'—as **PrU**.

3.4 Other Baselines

- Random prediction: we count label distribution from the training set and assign labels to test examples based on the obtained distribution.

- Sentence classification: we take the original text sequence as input and use the hidden states corresponding to the [CLS] token as the relation representation. Since no entity information is provided, the encoded relation representation is sub-optimal. However, the model may still learn heuristics that the sentence mentions the relation (Rosenman et al., 2020). It is also worth noting that if multiple relations exist in the sentence, it is impossible for this baseline model to distinguish between

	DDI 2013	i2b2-2010	ReTACRED	SciERC	TACRED	RadGraph
# entity pairs	31,784	65,210	91,467	4,648	106,264	25,848
no-relation pairs	84.5%	85.6%	63.2%	0.0%	79.5%	50.0%
# relations	5	9	40	7	42	4
# entity types	4	3	16	6	17	4
Avg # tokens per sentence	27.9	19.5	36.3	25.5	36.4	111.4
Avg # entities per sentence	4.6	4.5	2.2	7.4	2.1	29.2
Avg # tokens per entity	1.3	2.4	1.6	2.4	1.6	1.0
Avg # tokens btw pairs	15.4	12.2	12.0	6.2	12.1	2.7

Table 2: The descriptive statistics of the datasets.

them.

- Entity name only: we keep two target entities and remove all other tokens. Taking the sentence in Figure 1 as an example, the sentence becomes ‘Rx cramps’ and is fed as input to the sentence classifier.

- Entity type only: instead of entity names, we use entity type and remove all other tokens. The example sentence in Figure 1 becomes ‘Treatment Problem’.

4 Datasets and Experimental Setup

We choose six datasets—all in English—that are sampled from four different domains:

DDI 2013 (Segura-Bedmar et al., 2013) is sampled from biomedical publications. Four entity types—*brand*, *drug*, *drug_n* and *group*—and four relation types—*advise*, *effect*, *int*, and *mechanism*—are annotated in the dataset.

i2b2-2010 (Uzuner et al., 2011) is sampled from clinical notes. Three entity types—*test*, *problem* and *treatment*—and eight relation types—*TrWP*, *TrNAP*, *TeCP*, *TrCP*, *TrIP*, *TrAP*, *TeRP* and *PIP*—are annotated in the dataset.

TACRED and ReTACRED by (Zhang et al., 2017) and (Stoica et al., 2021) are sampled from newswire and the web. Forty-one relations, such as *per:date_of_birth* and *org:shareholders* exist in TACRED. The original relation labels of TACRED are obtained by crowd-sourcing, and the later work (Alt et al., 2020; Stoica et al., 2021) show that the quality of crowd-sourced annotations is a major factor contributing to the overall error rate of models on TACRED. Therefore, we use both TACRED and ReTACRED, a label-corrected version released by Stoica et al. (2021).

SciERC (Luan et al., 2018) is sampled from computer science publications. Six entity types—*Method*, *Generic*, *Material*, *Task*, *Metric* and *OtherScientificTerm*—and seven relation types—*USED-FOR*, *EVALUATE-FOR*, *CONJUNCTION*, *HYPONYM-OF*, *FEATURE-OF*, *COMPARE* and *PART-OF*—are annotated in the dataset.

RadGraph (Jain et al., 2021) is sampled from clinical notes. Radiology reports are annotated with four types of entities: Anatomy, Definitely Present Observation, Uncertain Observation, and Definitely Absent Observation; and three types of relations: Suggestive Of, Located At, and Modify.

The descriptive statistics of the datasets are listed in Table 2. On TACRED, ReTACRED and RadGraph, we use the official train-dev-test split. We use the split of SciERC from Gururangan et al. (2020); both DDI 2013 and i2b2-2010 from the BLUE benchmark (Peng et al., 2019).

We use RoBERTa-large as the backbone model in all our experiments except for the prompt-based models. We use BART-large in prompt-based experiments because BART is an encoder-decoder model that is pre-trained by reconstructing the original text from the corrupting text.

For each model variant, we fine-tune the whole model and perform a grid search to find the best combination of the number of training epochs and the learning rate on each development set. Once the best combination is found, we repeat all experiments three times using different random seeds, and medium test Micro and Macro F_1 scores are reported. In addition to evaluation results on all test examples, we follow (Zhou and Chen, 2022) and report results on filtered test sets, where test examples containing entities observed in the training set are removed.

	Method	Dataset						
		DDI 2013	I2B2-2010	RETACRED	SCIERC	TACRED	RADGRAPH	AVG
	random prediction	5.5/6.6	3.6/2.7	4.7/1.4	28.1/12.7	1.5/0.8	22.3/14.5	10.9/6.5
	w/o entity info	41.5/35.4	48.1/37.8	42.8/28.9	60.9/43.0	24.1/20.6	1.7/0.9	36.5/27.8
	entity name only	13.8/10.0	63.9/39.8	65.8/31.8	67.9/53.7	47.2/24.6	92.8/90.3	58.6/41.7
	entity type only	4.2/3.4	48.1/15.0	56.2/19.3	61.5/32.6	33.1/8.7	74.2/78.6	46.2/26.3
SpU	Span-based	82.6/78.2	77.5/63.9	88.1/76.6	87.9/81.6	68.0/53.1	94.0/95.3	83.0/74.8
SpT	+ entity type	82.6/77.1	77.7/66.5	88.8/75.6	88.4/81.5	68.3/52.9	94.2/94.5	83.3/74.7
MaU	Marker-based	**84.2**/79.0	80.4/69.9	90.5/80.5	88.8/81.5	70.2/55.2	93.8/93.3	84.7/76.6
MaTi	+ entity type (insert)	**84.2**/**80.3**	**82.2**/**71.1**	**90.7**/82.0	89.8/83.9	**73.9**/**60.6**	**95.9**/**96.3**	**86.1**/**79.0**
MaTr	+ entity type (replace)	83.6/80.1	80.1/70.6	86.5/75.6	88.2/82.0	71.9/56.1	92.6/94.0	83.8/76.4
PrU	Prompt-based	74.8/70.6	74.3/61.3	90.2/81.9	89.9/**84.9**	71.1/55.9	93.1/94.2	82.2/74.8
PrT	+ entity type	75.2/71.8	77.1/66.5	90.6/**84.3**	**90.0**/84.6	73.5/60.4	95.2/96.1	83.6/77.3

Table 3: A comparison of methods incorporating entity type information to the RC model. Both Micro F1 and Macro F1 are reported. Underlined results indicate the improvement due to the incorporation of entity types is statistically significant (Wilcoxon signed-rank test, $p < 0.05$). The best Micro and Macro F1 results for each dataset are **boldfaced**.

	SpU	SpT	MaU	MaTi	PrU	PrT
DDI 2013 (479)	80.0	81.0	81.8	**82.1**	73.3	78.7
I2B2-2010 (10132)	74.4	75.2	77.1	**80.3**	71.3	74.8
RETACRED (3736)	87.9	87.5	90.4	**91.1**	90.1	90.8
SCIERC (561)	88.2	88.9	88.6	90.0	**90.2**	89.8
TACRED (4470)	72.7	72.5	74.6	**78.5**	74.4	77.8
RADGRAPH (44)	71.8	76.2	71.4	**81.8**	68.2	**81.8**
AVG	79.2	80.2	80.6	**84.0**	77.9	82.3

Table 4: Micro F_1 results on filtered test sets, where examples containing seen entities from the training sets are removed. Numbers in parentheses are the number of examples in the filtered test sets.

5 Results and Analysis

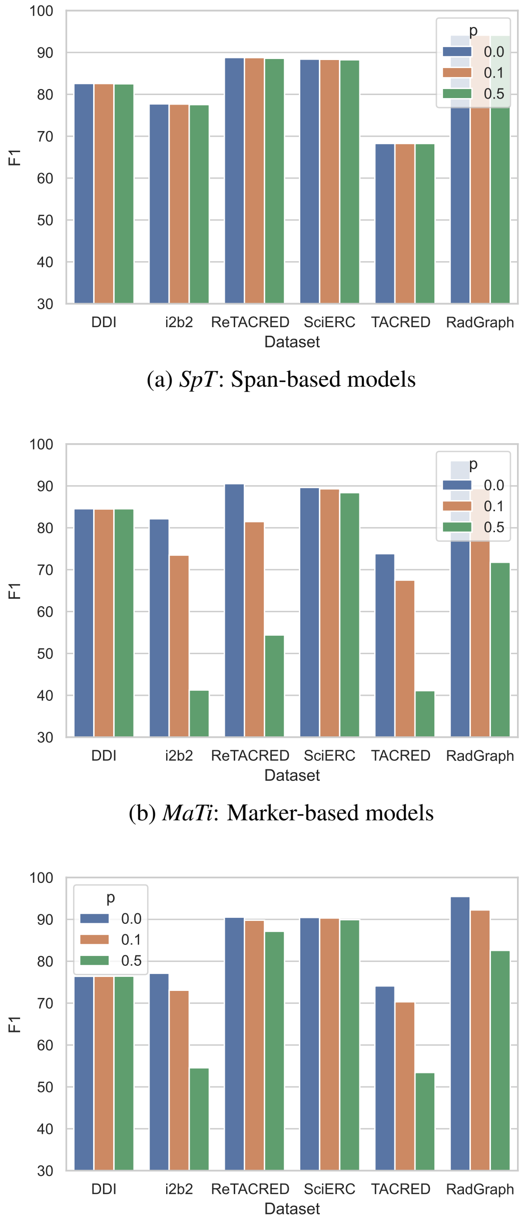
The first observation from Table 3 is that incorporating gold entity type information can improve the effectiveness of marker-based models. On four out of six datasets, inserting typed makers (*MaTi*) significantly outperforms using untyped markers (*MaU*) in terms of Micro F_1, and the averaged improvement of Micro F_1 over all datasets is 1.4 (Macro of 2.4). Similarly, the averaged improvement of Micro F_1 using entity type information with prompt-based models is 0.6 and span-based is 0.3 (the averaged Macro F_1 of span-based models slightly decreases 0.1 when entity type information is incorporated). Secondly, we observe that marker-based models outperform span-based and prompt-based models on most of these evaluations, except on RETACRED and SCIERC, where prompt-based models achieve the highest Micro or Macro F_1. Thirdly, replacing entity names using typed markers (*MaTr*) under-performs inserting typed markers before and after entity names (*MaTi*) with a large margin (on average 2.3 Micro F_1). Finally, we observe that the benefits of incorporating entity type information seem to be dataset dependent. For example, *MaTi* significantly outperforms *MaU* on SCIERC, TACRED, and RADGRAPH in terms of both Micro and Macro F_1 scores. *PrT* significantly outperforms *PrU* on I2B2-2010, RETACRED, TACRED, and RADGRAPH in terms of both F_1 scores.

When RC models are evaluated on examples containing unseen entities (Table 4), we can see incorporating entity type information brings larger improvements compared to results on the complete test set (on average 1.0 vs 0.3 with span-based; 3.4 vs 1.4 with marker-based; and 4.4 vs 1.4 with prompt-based models). This result shows that using entity type information improves the generalisation of the RC models to unseen entities.

5.1 Effectiveness Drop due to Incorrect Entity Type Information

After we investigate the benefits of incorporating gold entity type information to the RC model, the next question is: what will happen if incorrect entity type information is used during inference? We believe that gold (human-annotated) entity type information may be available on a small scale and can be used to train the RC model. However, it is impractical to expect entity type information to be always correct when the RC model is employed in the wild. Therefore, we focus on analysing the *robustness* of RC models against incorrect entity types and measuring how the effectiveness of trained RC models is affected by the correctness of entity type information during inference.

For each target entity in the test example, we



(a) *SpT*: Span-based models

(b) *MaTi*: Marker-based models

(c) *PrT*: Prompt-based models

Figure 3: (a): negligible effectiveness drop due to incorrect entity types using span-based models; (b): large drop on some (but not all) datasets using marker-based models; (c): moderate drop using prompt-based models. High p values indicate more entity type errors.

use a binomial distribution, $p \in [0, 1]$, to randomly decide whether its entity type should be corrupted. If yes, we select another entity type—the incorrect type with the highest output probability based on a span-based NER model—as the replacement. Note that we also investigate randomly sampling erroneous entity types and observe a pattern similar to NER-based errors.

We train span-based NER models (Zhong and Chen, 2021; Dai and Karimi, 2022) separately on the corresponding training sets using entity annotations. The model enumerates all possible spans and determines whether a span is a valid entity and its entity type. The accuracy—given an entity name in the context, predict its entity type—of these trained NER models are high on DDI 2013 (93.4), i2b2-2010 (91.4), RadGraph (91.7) and relatively low on ReTACRED (71.7), TACRED (71.4), SciERC (71.6). Two possible factors are causing this accuracy divergence. On the one hand, this difference reflects that classifying entity names of different types has various levels of inherent difficulty (e.g., it may be easy to identify drug names in DDI 2013, but difficult to identify the metric names in SciERC). On the other hand, the low accuracy on some datasets can be attributed to the scarcity of entity annotations or the noisy annotations (e.g., entity names in ReTACRED and TACRED are not fully annotated).

The sensitivity analysis results show that span-based models are robust against incorrect entity types (Figure 3a). When entity types are incorrect, the RC models still maintain similar effectiveness as the gold entity types used. In contrast, incorrect entity types cause large effectiveness drop on some (but not all) datasets with mark-based models (Figure 3b) and moderate drop with prompt-based models (Figure 3c). For example, when 10% of entity types are incorrect ($p = 0.1$), marker-based models have had great effectiveness drop on i2b2-2010 (8.7), ReTACRED (9.1), TACRED (6.3), and RadGraph (6.7). We argue this result shows state-of-the-art models (Zhou and Chen, 2022)—inserting typed markers before and after entity names—to be a questionable design option in practice, although they indeed achieve the highest F_1 scores on most of the evaluations when gold entity type information is used. It is also worth noting that on DDI 2013 and SciERC, even when 50% of target entities have incorrect entity types, the drop of mark-based models is still very small (0.0 and 1.2, respectively).

5.2 What can associations between relation and entity types tell us?

To understand why on some, but not all, datasets marker-based and prompt-based models have performance drop using incorrect entity types, we use Pointwise Mutual Information (PMI), an association measure to quantify the strength of associ-

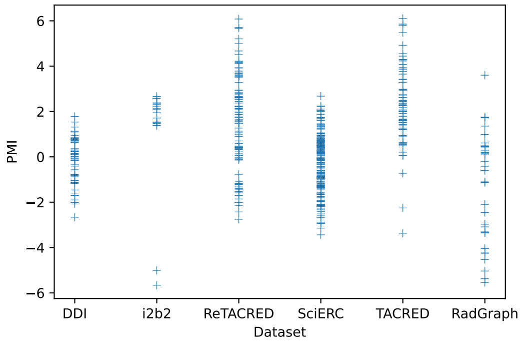


Figure 4: Association between relation (e.g., ‘TrAP’) and entity type pair (e.g., (‘Treatment’, ‘Problem’)), measured using PMI values, on different datasets.

ation between relation and entity types. Taking the sentence in Figure 1 as an example, we denote the relation ‘TrAP’ as r and a pair of entity types (‘Treatment’, ‘Problem’) as e. We calculate PMI by considering the number of occurrences in the training set:

$$\text{PMI}(r, e) = \log \frac{\#(r, e) \times |D|}{\#(r) \times \#(e)}, \tag{1}$$

where $|D|$ is the total number of examples, $\#(r)$ is the frequency of relation, $\#(e)$ is the frequency of entity type pair, $\#(r, e)$ is the frequency r and e occur.

The measured PMI values on different datasets are shown in Figure 4. On DDI 2013 and SciERC, possible combinations of relation and entity type pairs are more evenly distributed and centred at zero, indicating the strength of association on these datasets is weak. Therefore, even if a large portion of incorrect entity types are provided, the RC model is still able to make the correct prediction (see the negligible drop in Figure 3). In contrast, values of other datasets have more imbalance distribution across a larger range. It indicates that relation types have stronger—either positive or negative—association with entity type pairs. Therefore, if incorrect entity types are provided, the RC model is more likely to make a wrong prediction (see the large drop in Figure 3b and Figure 3c).

5.3 A closer look at the examples

We provide a few representative examples in this section to demonstrate how (incorrect) entity type information might affect the effectiveness of RC models:

- **The correct entity type information helps the relation prediction.** For example, given the sentence taken from the SciERC dataset, *‘Hitherto , smooth motion has been encouraged using a trajectory basis , yielding a hard combinatorial problem with time complexity growing exponentially in the number of frames .’*, the marker-based approach (*MaU*) predicts the relation between *‘time complexity’* and *‘hard combinatorial problem’* is ‘FEATURE-OF’ when no entity type information used. This prediction is likely to be influenced by the preposition *‘with’* between these two entity mentions. However, once the correct entity type information (‘Metric’ and ‘Task’) is given, the model (*MaTi*) correctly predicts the ‘EVALUATE-FOR’ relation.

- **The incorrect entity type information causes erroneous predictions, whereas models without using entity type and models using correct entity type succeed.** For example, given the sentence taken from the TACRED dataset, *‘The troubled insurance giant , which has received multiple federal bailouts since September , said that it would give the New York Fed preferred stakes in two of the company ’s crown jewels Asian-based American International Assurance , or AIA , and American Life Insurance Co. , or Alico , which operates in more than 50 countries .’*, both the model without using entity type information and the one using gold entity type information can predict correctly the relation between *‘Alico’* and *‘American Life Insurance Co.’* is ‘org:alternate_names’. However, when the NER model makes a mistake and recognises ‘Alico’ as a person name, the relation model is negatively affected and predicts the relation ‘org:top_members/employees’, a common relation between a persona and an organisation.

- **Incorrect entity type information does not cause erroneous relation predictions.** For example, given the sentence taken from the SciERC dataset, *‘Amorph recognises NE items in two stages : dictionary lookup and rule application .’*, models with incorrect entity type—NER model predicts both *‘dictionary lookup’* and *‘rule application’* as ‘Task’ instead of ‘Method’—can still predict the relation between *‘dictionary lookup’* and *‘rule ap-*

plication' as 'conjunction' relation due to the existence of the conjunction between them.

5.4 Implications

RC models are usually employed as subcomponents of IE systems, and entity type information is generated using an automated NER system. Depending on the effectiveness of the NER and the association between relation type and entity type pairs, we suggest using different RC variants. If the association between relation and entity type pairs is weak (e.g., DDI 2013, SciERC), we suggest using the marker-based model with entity type information used. If relation types have a strong association with entity types (e.g., i2b2-2010, RadGraph, TACRED, ReTACRED), we suggest choosing prompt-based models if relatively accurate entity types are guaranteed or span-based models if entity types are prone to errors.

6 Conclusions

Relation Classification (RC) is an active area of research for a number of applications, such as knowledge base construction and biomedical text mining. The existing methods often heavily rely on entity information, especially entity type information. We conduct a comparison of methods of incorporating entity type information into the RC models on six datasets across four different domains. Results show that when gold entity type information is available, inserting typed markers before and after target entities and using token representations corresponding to these typed markers for relation representation is effective. However, when entity types become inaccurate, methods that rely on typed markers become less effective on some (but not all) datasets. In contrast, span-based methods that use token representations to build span representation and then relation representation are robust when incorrect entity types are provided. The latter is a more realistic scenario, given NER models are practically never perfect. The prompt-based method that uses entity type information as part of the initial decoder inputs is located in the middle of the spectrum. It is also affected by the incorrect entity types, but its performance drop is much smaller than the one with marker-based models.

We found that Pointwise Mutual Information, a measure to quantify the association between relation and entity type pairs, can explain why on some datasets entity type errors cause large effectiveness drops. We suggest it as a cheap yet effective tool to understand the dataset and help the decision about how to use entity type for the RC model.

Limitations

Our work is motivated by debates on the usefulness of entity type information for relation classification. We investigated how the effectiveness of relation classification models is affected by the correctness of entity type information. However, the effectiveness of relation classification models can be affected by other factors, such as entity names (whether the NER model can effectively identify entity boundaries) and surrounding context (whether there is sufficient context). We leave the investigation of other factors for future work.

Ethics Statement

There are no known ethical concerns associated with the findings of this work. However, we acknowledge that all datasets used are in English, which does not help mitigate the inequality of NLP research across languages.

Acknowledgement This work is supported by the CSIRO's Precision Health Future Science Platform (PH FSP). This work is also undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government.

References

Heike Adel and Jannik Strötgen. 2021. Enriched Attention for Robust Relation Extraction. *arXiv*, 2104.10899.

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task. In *ACL*.

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Improving Relation Extraction by Pre-trained Language Representations. In *AKBC*.

Ivan Bilan and Benjamin Roth. 2018. Position-aware Self-attention with Relative Positional Encodings for Slot Filling. *arXiv*, 1807.03052.

Yuxuan Chen, David Harbecke, and Leonhard Hennig. 2022. Multilingual Relation Classification via Efficient and Effective Prompting. In *EMNLP*.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint Extraction of Entities and Relations for Opinion Recognition. In *EMNLP*.

Mark Craven and Johan Kumlien. 1999. Constructing Biological Knowledge Bases by Extracting Information from Text Sources. In *ISMB*.

Xiang Dai and Sarvnaz Karimi. 2022. Detecting Entities in the Astrophysics Literature: A Comparison of Word-based and Span-based Entity Recognition Methods. In *WIESP@AACL*.

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

Cícero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying Relations by Ranking with Convolutional Neural Networks. In *ACL-IJCNLP*.

Markus Eberts and Adrian Ulges. 2020. Span-based Joint Entity and Relation Extraction with Transformer Pre-training. In *ECAI*.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *ACL*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *ACL*.

Jiale Han, Shuai Zhao, Bo Cheng, Shengkun Ma, and Wei Lu. 2022. Generative Prompt Tuning for Relation Classification. In *Findings of EMNLP*.

Saahil Jain, Ashwin Agrawal, Adriel Saporta, Steven Q H Truong, Du Nguyen Duong, Tan Bui, Pierre Chambon, Yuhao Zhang, Matthew P Lungren, Andrew Y Ng, Curtis P Langlotz, and Pranav Rajpurkar. 2021. RadGraph: Extracting Clinical Entities and Relations from Radiology Reports. In *NeurIPS*.

Heng Ji and Ralph Grishman. 2011. Knowledge Base Population: Successful Approaches and Challenges. In *ACL*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *TACL*, 8.

Yang Li, Guodong Long, Tao Shen, Tianyi Zhou, Lina Yao, Huan Huo, and Jing Jiang. 2020. Self-attention enhanced selective gate with entity-aware embedding for distantly supervised relation extraction. In *AAAI*.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2021. EntityBERT: Entity-centric Masking Strategy for Model Pretraining for the Clinical Domain. In *BioNLP@NAACL*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A Joint Neural Model for Information Extraction with Global Features. In *ACL*.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *EMNLP*.

Shengfei Lyu and Huanhuan Chen. 2021. Relation Classification with Entity Type Restriction. In *Findings of ACL*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL-AFNLP*.

Makoto Miwa and Mohit Bansal. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *ACL*.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2015. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104.

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. In *EMNLP*.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In *BioNLP@ACL*.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. In *EMNLP-IJCNLP*.

Shachar Rosenman, Alon Jacovi, and Yoav Goldberg. 2020. Exposing Shallow Heuristics of Relation Extraction Models with Challenge Data. In *EMNLP*.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In *SemEval*.

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

Daniil Sorokin and Iryna Gurevych. 2017. Context-Aware Representations for Knowledge Base Relation Extraction. In *EMNLP*.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-TACRED: Addressing Shortcomings of the TACRED Dataset. In *AAAI*.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *JAMIA*, 18.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. 2016. Combining Recurrent and Convolutional Neural Networks for Relation Classification. In *NAACL*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of ACL*.

Qiang Wei, Zongcheng Ji, Yuqi Si, Jingcheng Du, Jingqi Wang, Firat Tiryaki, Stephen Wu, Cui Tao, Kirk Roberts, and Hua Xu. 2019. Relation Extraction from Clinical Narratives Using Pre-trained Language Models. In *AMIA*.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *CIKM*.

Xin Xu, Xiang Chen, Ningyu Zhang, Xin Xie, Xi Chen, and Huajun Chen. 2022. Towards Realistic Low-resource Relation Extraction: A Benchmark with Empirical Baseline Study. In *Findings of EMNLP*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *EMNLP*.

Zhaohui Yan, Zixia Jia, and Kewei Tu. 2022. An Empirical Study of Pipeline vs. Joint Approaches to Entity and Relation Extraction. In *AACL*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named Entity Recognition as Dependency Parsing. In *ACL*.

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

Dongxu Zhang, Subhabrata Mukherjee, Colin Lockard, Luna Dong, and Andrew McCallum. 2019a. OpenKI: Integrating Open Information Extraction and Knowledge Bases with Relation Inference. In *NAACL*.

Dongxu Zhang and Dong Wang. 2015. Relation Classification via Recurrent Neural Network. *arXiv*, 1508.01006.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *EMNLP*.

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

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019b. ERNIE: Enhanced Language Representation with Informative Entities. In *ACL*.

Zexuan Zhong and Danqi Chen. 2021. A Frustratingly Easy Approach for Entity and Relation Extraction. In *NAACL*.

Wenxuan Zhou and Muhao Chen. 2022. An Improved Baseline for Sentence-level Relation Extraction. In *AACL-IJCNLP*.

A Resources and Downloads

TACRED `https://catalog.ldc.upenn.edu/LDC2018T24`

ReTACRED `https://github.com/gstoica27/Re-TACRED`

RadGraph `https://physionet.org/content/radgraph/1.0.0/`

SciERC `https://github.com/allenai/dont-stop-pretraining`

DDI 2013 `https://github.com/ncbi-nlp/BLUE_Benchmark`

i2b2-2010 `https://github.com/ncbi-nlp/BLUE_Benchmark`

RoBERTa-large `https://huggingface.co/roberta-large`

BART-large `https://huggingface.co/facebook/bart-large`

GenPT `https://github.com/hanjiale/GenPT`