# Relation Extraction with Data Augmentation

**Haoliang Jiang, Yingjun Mou, Shangming Zhao and Fan Bai**
Georgia Institute of Technology
{hjiang321, ymou32, szhao332, fbai31}@gatech.edu

## Abstract

The state-of-the-art relation extraction (RE) models can achieve great performances on the benchmark datasets. However, they have been found to rely on shallow heuristics such as entity type when making predictions instead of learning the semantic relation between two given entities. In this paper, we tackle the heuristic-reliance problem in SOTA RE models using a data augmentation method. We generate synthetic relation extraction data based on the co-reference information between different entity mentions, and then use the generated data to boost our training data. As a result, the SOTA model trained on synthetic data achieves much better performance on the challenging relation extraction (CRE) dataset while maintaining its performance on the benchmark TACRED, showing the effectiveness of our data augmentation strategy.

## 1 Introduction

Relation extraction (RE) is a long-standing problem in the field of Information Extraction (IE). Its standard setup is: given an input sentence $s$ and a set of mentioned entities $e \in E$, we want to extract all possible tuples $(e_1, e_2, r, s)$, where $r$ holds between $e_1$ and $e_2$. Recent work tends to frame RE as a multi-class classification problem, known as relation classification (RC), in which only two entity mentions in the sentence are considered, and the model needs to predict a relation $r$, from a set of pre-defined relations $R \cup \emptyset$. The popular benchmark datasets TACRED (Zhang et al., 2017) was annotated using this RC framework, where only 17.2% entity pairs are annotated.

From the model side, to tackle this RC setup, what the state-of-the-art (SOTA) model does is to generate the contextual representations of two input entities using a pre-trained language model like BERT (Devlin et al., 2019), and then concatenate them as the sentence representation and the input to a final classification layer. Although this simple architecture leads to a competitive performance, the resulting RE model tends to have a "right for the wrong reasons" issue (McCoy et al., 2019). According to Rosenman et al. (2020), the SOTA model performs RE relying on three shallow heuristics when trained on TACRED: 1) *entity types*; 2) *existence of an event without linking the event to its arguments*; 3) *combination of first two heuristics (event + type)*. The authors show that while those heuristics are not well attested in the dev and test sets of TACRED, they can hurt the model performance on challenging relation extraction (CRE) examples, a self-built dataset to address the heuristic-relying problem.

In this paper, we tackle the heuristic-relying issue in the current RE models from a data augmentation perspective as we believe that annotating more entity pairs in the same sentence can help the RE model understand the interaction between different entities better and alleviate its reliance on the surface patterns without connecting events to specific arguments. Because manual annotation is expensive, we developed a rule-based system to generate synthetic data to boost our training data. As introduced previously, in RC setup, each example comes with two entities, so we first detect all the named entity mentions in the input sequence using off-the-shelf NLP tools/models. Then we filter out unrelated entities and use the remaining ones to generate synthetic labeled data using two following assumptions:

- Replacing an entity in the given entity pair with any of its co-reference mentions in the sentence does not change the relation.
- Replacing an entity in the given entity pair with an non-co-reference entity changes the original positive relation to "no_relation".

Using the above two assumptions, we generated synthetic examples using the training set of TACRED. We then combined them with the original

training data and re-trained the RE model. We evaluated our re-trained model on both TACRED dataset and the CRE dataset. We find that adding the synthetic data can greatly improve the performance on CRE, showing the effectiveness of addressing the heuristic-relying issue in the state-of-the art RE model, while maintaining its performance on TACRED.

## 2 Method

In this section, we show how we address the heuristics-reliance problem in RC models using data augmentation. We start with detecting all the mentions in the input sentence in §2.1. Then we describe how we generated the synthetic RE data using detected entities and original gold relation under two important assumptions in §2.2.

### 2.1 Entity Detection

#### 2.1.1 Named Entity Recognition

The first step to generate synthetic RE data is to find all possible mentions in the input sequence. Here we use the NER module of Stanford CoreNLP, a CRF-based tagger (Finkel et al., 2005) plus multiple rule-based systems (Chang and Manning, 2012), because its output entity types align with the entity types of two RE datasets TACRED and CRE we want to experiment on. To have a sense of how reliable our deployed NER system is, we calculated the recall of two given entities in the RE datasets, and the results are shown in Table 1. We can see that the recall scores are high in general for both datasets (>0.88) while TACRED has a slightly worse performance compared to CRE mainly because its entities are labeled by human annotators.

| Dataset | Recall |
|---|---|
| TACRED-train | 0.888 |
| TACRED-dev | 0.893 |
| TACRED-test | 0.883 |
| CRE | 0.940 |

Table 1: Recall of two given entities in RE datasets.

#### 2.1.2 Co-reference Resolution

Once we have all the entity mentions in the sentence, the next step is to run a co-reference model to see whether difference mention are co-referred to the same entity. This information is very importance for later synthetic data generation. We used three different frameworks to extract co-reference clusters from the RE dataset: a statistical model from Stanford CoreNLP, a neural net scoring tool called neuralcoref, and the c2f-coref + SpanBERT model. In all of the three models, we considered each sample's text as a independent document.

**CoreNLP** We used the statistical co-reference model from CoreNLP, this model uses a large set of features to perform mention-ranking on the document (Manning et al., 2014).

**NeuralCoref** This model is based on the neural scoring model described in (Clark and Manning, 2016).

**c2f-coref + SpanBERT** c2f-coref + SpanBERT is a span-based co-reference model which learns a distribution $P(\cdot)$ over possible antecedent spans for each mention span, and uses SpanBERT for contextual encoder. We used the model trained on OntoNotes (Weischedel et al., 2013) to extract co-reference clusters from RE datasets.

**Comparing Performances** Similarly, to estimate how we the off-the-shelf models perform on our RE dataset, we randomly sampled 30 examples from the TACRED dataset, and manually labeled co-reference clusters for each sample, and used these samples to evaluate the performance of our co-reference models. Similar to (Joshi et al., 2020), we used 3 evaluation metrics: MUC, $B^3$, and $CEAF_{\phi_4}$. The results are shown in table 2.

| | MUC F1 | $B^3$ F1 | $CEAF_{\phi_4}$ F1 | Avg. F1 |
|---|---|---|---|---|
| CoreNLP | 0.427 | 0.450 | 0.495 | 0.457 |
| Neuralcoref | 0.393 | 0.397 | 0.415 | 0.402 |
| SpanBERT | **0.507** | **0.520** | **0.573** | **0.533** |

Table 2: Co-reference evaluation results on 30 manually labelled random samples

The SpanBERT significantly outperfomed the other 2 methods, but its average F1 score (0.533) is much lower than that of SpanBERT on OntoNotes (0.796). After looking at the predictions, we found that SpanBERT's outputs make sense at most of the time. The low score is mainly caused by the following reasons:

1. The nature of the evaluation metrics. Since the 3 metrics are all based on set operations, and most of the samples contain only 1-2 clusters, a single mismatch between two mentions can make the F1 of a sample drop by 30%. This is the main reason of performance drop.

2. Different labelling conventions. For example, in the sample “Kissel" is labelled as a mention, while in SpanBERT “Kissel ’ s" is labelled together as a mention. We suspect that this is because the SpanBERT is trained on another dataset, the OntoNotes, where the labelling conventions might differ from what we used.

3. Reasonable but imprecise labelling. For example, in the sample “Claude - Alain Margelisch" and “CEO of the Swiss Bankers Association" are considered as two different mentions, while in SpanBERT these two are concatenated as one single mention.

Therefore we conclude that the co-reference predictions from SpantBERT do provide meaningful information, and can be used in our relation extraction model. It got low score in terms of the evaluation metrics because some of the spans it predicted didn’t exactly match with that in our labelled samples, though these spans referred to meaningful mentions. At this point, we didn’t find fuzzy matching algorithms or better metrics to handle this problem.

### 2.2 Synthetic Data Generation

To generate synthetic RE data, given an input sentence and two entities, we first clean the entity mentions obtained from NER (§2.1.1) and only keep mentions which have the same type with (one of) two given entities. Then based on the co-reference information (§2.1.2), we classified the remaining entities into two collections depending on whether an entity is the co-reference mention (in the same co-reference cluster) of one of the given entities. If yes, it will be classified to the first collection, otherwise the second one.

Once we have all the entity mentions classified, the next step is to use them to generate synthetic RE training data. We make two assumptions for synthetic data generation:

- Replacing an entity in the given entity pair with any of its co-reference mentions in the sentence does not change the relation.

- Replacing an entity in the given entity pair with an non-co-reference entity (same entity type) can change the original positive relation to "no_relation".

The first assumption is based on the essence of co-reference, and we use it for the entity mentions in the first collection to generate synthetic RE data. The second assumption is based on the exclusivity of relational facts. For example, if there is a "title_of" relation between two given entities (one "PERSON" entity and one "TITLE" entity), it is highly likely that all the other "PERSON" entities in the same sentence have "no_relation" with the given "TITLE" entity. This assumption can be used for the entities in the second collection to generate challenging negative data because remembering surface patterns does not work here and thus address the heuristic-relying problem.

## 3 Relation Extraction

In this section, we introduce two styles of relation extraction models used in our experiments: 1) sentence-level RE, which predicts the relation between two given entity mentions; 2) document-level RE, which predict the relation between two given entities considering all the mentions in the input sequence.

### 3.1 Sentence-level RE

We experiment two sentence-level RE architectures. The first architecture follows (Soares et al., 2019), where four special tokens `[E1]`, `[\E1]`, `[E2]`, `[\E2]` are used to specify the positions of two given entities, and the contextualized embeddings of `[E1]` and `[E2]` are concatenated as input to a final linear layer that was used to predict the relation. For the second architecture, it is similar to the first one with the only difference that the entity type information is encoded into the four special tokens (Tamari et al., 2021; Zhong and Chen, 2020). For example, when the types of the subject and the object are `PERSON` and `LOCATION` respectively, the four above special tokens become `[E1-PERSON]`, `[\E1-PERSON]`, `[E2-LOCATION]` and `[\E2-LOCATION]`.

### 3.2 Document-level RE

Due to our entity detection process (§2.1), we now have multiple mentions of one given entity , which is similar to the setup of document-level relation extraction setup, so we also experiment with the SOTA model on DOCRED (Yao et al., 2019) proposed by Zhou et al. (2020). This model integrates BERT with two novel modules, adaptive thresholding and localized context pooling. The adaptive thresholding model is to learn a threshold for identifying positive relations with the final confident

scores generated by the model. The method overcome the problem for document relation extraction as several golden relations exist in each sample of DOCRED. We remove this module in our experiment, as TACRED has only one golden relation for each sample. In localized context pooling (LCP), the attention heads are used directly to generate context embedding related to both entities. The setting of the model makes it possible to be used in a single-mention scenario like TACRED though that the original paper is targeting DOCRED.

## 4 Experiments

### 4.1 Data

We experiment two RE datasets. The first one is TACRED (Zhang et al., 2017), which contains 106,264 samples where each sample is labeled with a golden relation between two given entities. The samples are collected over newswire and web text from the corpus in TAC Knowledge Base Population challenges (Getman et al., 2018). There are totally 42 relations (41 positive relations plus a "$no\_relation$"), and around 80% of data are labeled as "$no\_relation$". We used the training set of TACRED to generate synthetic data and keep the dev and test sets unchanged.

The second dataset is CRE, which is collected to benchmark the limitation in the data annotation process of TACRED (Rosenman et al., 2020) and used only for evaluation. The dataset contains 10,844 instances in a similar format to TACRED. As mentioned previously, CRE argues about the $event + type$ heuristic for current SOTA relation extraction models. The dataset is a collection of samples that related to this specific failure which is meant only for evaluation and not training. The difference between TACRED and CRE lies on that CRE contains pairs that share the same events or types with the positive relation pairs for the same corpus which help to identify how the trained model relies on the $event + type$ heuristic. The original problem of relation extraction can be formed as $(e_1, e_2, s) \rightarrow r$ where $e_1$ and $e_2$ are denoted as entities; s is denoted as the sentence and r is denoted as the relation. CRE forms the problem into $(e_1, e_2, s, r) \rightarrow \{1, 0\}$ which is a binary classification problem. For each corpus, 3.7 candidate entity-pairs are collected on average.

### 4.2 Evaluation

The evaluation of TACRED utilizes the traditional metrics precision, recall and $F_1$ score. For CRE, it uses accuracy as the evaluation metric to show how the tested model utilize the $event + type$ heuristic. The total accuracy can also been separated into positive and negative accuracy. Positive accuracy demonstrates the tested model's performance in regular samples while negative accuracy shows the performance when the tested model is facing samples where entities of the same types as the positive samples but relation does not hold.

### 4.3 Implementation

We use the BERT-base for the contextual embeddings of all our models. Both sentence-RE and Document-RE models are implemented using Huggingface Transformers. We train our model for 5 epochs with the grid search in batch size among $\{4, 8, 16\}$ and learning rate among $\{1e-5, 2e-5, 5e-5\}$ on one GPU.

### 4.4 Model configurations

We experiment with following configureations:

**Sentence-RE** The BERT base model with linear classifier.

**Sentence-RE (entity type)** The Sentence-RE model with entity types as special tokens.

**Document-RE** The BERT base model using a linear classifier with data augmentation as mentions.

**Document-RE (entity type)** The Document-RE model with entity types as special tokens.

**Document-RE (LCP)** The Document-RE model with a LCP and a bilinear classifier.

## 5 Results & Analysis

Table 3 shows the performance for the models evaluated by TACRED and CRE. As we expect, utilizing entity type does help to improve the performance in TACRED. However, it degrades the performance of Sentence-RE model on CRE by an obvious gap when no data augmentation is used and improves the performance when data augmentation is used. In terms of the proposed data augmentation method, while it does not help much for vanilla Sentence-RE model, it boosts the performance for the Sentence-RE with entity types for both TACRED and the accuracy of CRE. We also observe

| Data | Model | TACRED | | | CRE | | |
|---|---|---|---|---|---|---|---|
| | | P | R | F1 | Acc | Acc+ | Acc- |
| TACRED | Sentence-RE | 69.3 | 67.4 | 68.3 | 73.4 | 78.1 | 69.6 |
| | Sentence-RE (entity type) | 74.7 | 66.6 | 70.4 | 70.4 | 77.6 | 64.6 |
| TACRED+Augmentation | Sentence-RE | 67.2 | 68.5 | 67.8 | 73.4 | 73.2 | 73.5 |
| | Sentence-RE (entity type) | 69.6 | 72.6 | **71.1** | 73.4 | 77.2 | 70.4 |
| | Document-RE | 70.1 | 65 | 67.4 | 71.8 | 70.1 | 72.3 |
| | Document-RE (entity type) | 70.9 | 65.7 | 68.2 | 73.7 | 70.2 | 76.4 |
| | Document-RE (LCP) | 68.5 | 65 | 66.8 | 73.6 | 75.4 | 72.3 |

Table 3: The test results on TACRED and CRE.

a performance degradation on TACRED when applying data augmentation with the Document-RE format.

In general, the data augmentation method in the Sentence-RE format is helpful in improving the model's performance in recall rate on TACRED. More importantly, it boosts the performance in $Acc_-$ and shrinks the gap between $Acc_+$ and $Acc_-$. In the case of Sentence-RE, Document-RE and Document-RE (entity type), the model achieves a even higher accuracy in negative samples. As indicated by (Rosenman et al., 2020), the gap between $Acc_+$ and $Acc_-$ shows how the tested model relies on the $event + type$ heuristic for prediction. Thus, we show that our method can mitigate this limitation of relation extraction models while does not downgrade the performance on regular relation extraction dataset like TACRED.

## 6 Conclusion

In this paper, we tackle the heuristic-reliance problem in SOTA RE models using a data augmentation method. We generate synthetic relation extraction data based on the co-reference and NER information obtained from off-the-shelf NLP tools, and then use the generated data to boost our training data. The results show that the SOTA sentence-RE model achieves great improvement on the challenging relation extraction (CRE) dataset after trained on our synthetic data, showing the effectiveness of our data augmentation strategy.

## References


Angel X. Chang and Christopher Manning. 2012. Sutime: A library for recognizing and normalizing time expressions.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

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

Jenny R. Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling.

Jeremy Getman, Joe Ellis, Stephanie Strassel, Zhiyi Song, and Jennifer Tracey. 2018. Laying the groundwork for knowledge base population: Nine years of linguistic resources for tac kbp. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *ArXiv*, abs/1902.01007.

Shachar Rosenman, Alon Jacovi, and Yoav Goldberg. 2020. Exposing Shallow Heuristics of Relation Extraction Models with Challenge Data. In *Proceedings of the 2020 Conference on Empirical Methods*

*in Natural Language Processing (EMNLP)*, pages 3702–3710, Online. Association for Computational Linguistics.

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

Ronen Tamari, Fan Bai, Alan Ritter, and Gabriel Stanovsky. 2021. Process-level representation of scientific protocols with interactive annotation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

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 the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

Zexuan Zhong and Danqi Chen. 2020. A frustratingly easy approach for joint entity and relation extraction. *ArXiv*, abs/2010.12812.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2020. Document-level relation extraction with adaptive thresholding and localized context pooling. *arXiv preprint arXiv:2010.11304*.