TriBlank: Multiple Entity Blanking for Relation Learning

Kabir Walia
Department of Computer Science
Cornell University
ksw63@cornell.edu

William Ma
Department of Computer Science
Cornell University
wm274@cornell.edu

Abstract

Efforts in sentence and document level relation extraction (RE) have often depended on predefined schema or knowledge graphs (KG) to develop mappings from text to relations, or to develop extendable textual representations. However, due to variations in KGs, it becomes difficult to develop a general relation extractor that can perform well on arbitrary relations. Soares et al., 2019 [9] leverage recent work in contextual word representations to construct task-agnostic relation representations. In this work, we build on their "Matching the Blanks" (MTB) training methodology by replacing entities in sets of three in a given example and task our model to identify relations between pairs of blanked entities. We evaluate our model, TriBlank, on the DocRED [11] dataset. Our results showcase the difficulty of identifying semantic relations in the absence of contextual entities.

1 Introduction

Relation Extraction (RE) is the task of identifying semantic relations between groups of entities in text. Apart from a wide-range of industrial applications, this task often forms the basis for other natural language processing tasks like question answering (Li et al., 2019) [4], knowledge base development (Trisedya et al., 2019) [10] and other natural language understanding tasks. From feature and kernel based methods to recent neural approaches, there has been a strong dependence on structured schema or knowledge graphs (KG) that encode semantic relation information for triples of the form (e_1, e_2, r) where e_1 and e_2 are entities and r is the semantic relation between them. The major drawback to KG-based methods is the inconsistency between KGs. The same set of entities could have multiple relations, different relations, or even no relations at all depending on the KG. The large variance between such ontologies is a roadblock to building general relation extractors that are capable of identifying semantic relations between an arbitrary pair of entities.

[E0] **Chelsea** [/E0] was an early [E1] **1970s** [/E1] band from [E2] **New York City** [/E2] , best known for being the band of drummer Peter Criss before he joined Kiss . They released one album , the self - titled album Chelsea in 1971 and then collapsed during the recording of their unreleased second album . In August 1971 , the band became Lips (a trio consisting of Criss and his [E0] **Chelsea** [/E0] bandmates Michael Benvenga and Stan Penridge) .

Figure 1: Entities (bolded) with entity markers (red) from DocRED. Multiple references of the same entity (Chelsea) are identified by the same entity markers.

Soares et al. (2019) [9] propose a novel pre-training methodology that produces task-agnostic relation representations without any KG or human annotation. They leverage BERT (Devlin et al., 2018) [1] which learns context-aware word embeddings. They blank out the target pairs of entities during the training process and try to learn representations for the relation using contextual information. By further progressing Lin and Pantel (2001)'s [5] extension of the Harris distributional hypothesis (Harris, 1954) [2] to RE, they base their methods on the notion that if a pair of entities share a specific relation in one sentence, then any other sentence that includes those entities is likely to express that same relation.

The interesting aspect to consider here is that contexts around such entities might themselves involve other entity tokens. And these contextual entities might impact the relation between the target pair. For example, in Figure 1, if we were predicting the relation between target enti-

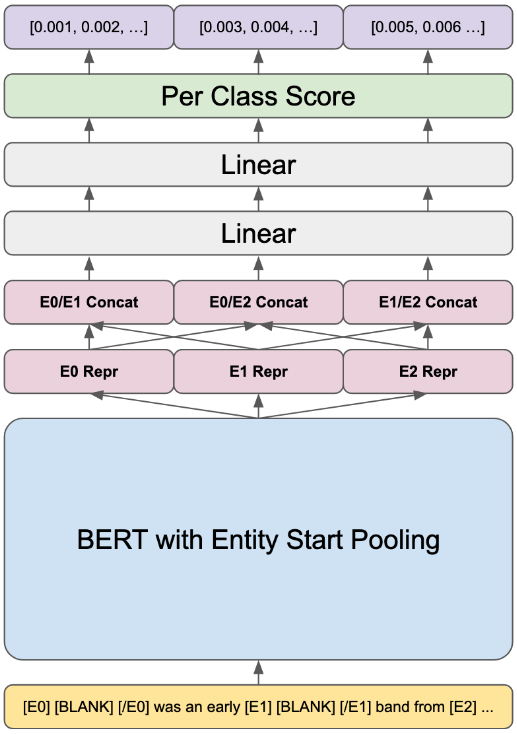

Figure 2: The TriBlank model architecture.

ties `[E0]` (Chelsea) and `[E2]` (Chelsea), `[E1]` (1970) would be considered as a contextual entity. It's possible that a BERT-based model (or any large pre-trained model) may identify a relation like "cities" or "locations" but adding in the "1970s" token might better contextualize the two words. Therefore, in this work we extend this blanking-based training methodology to three entities and task models to learn to predict pairwise relations without a third contextual entity. Our model architecture is called TRIBLANK, and we employ this training method on the DocRED [11] dataset — a widely used benchmark for document level RE tasks. Our results show that excluding contextual entity information makes relation extraction a harder task.

2 Related Works

The impact of contextual entities on semantic relations between target entities has been studied in previous works. Singh and Bhatia (2019) [8] explore the idea that two entities can be explicitly connected via a context token in a document. They find a token in the text that is most related to both target entities, and compute the a relation score between the two target entities as the summation of their relation scores with the contextual token. They achieve SOTA results on two biomedical datasets, indicating the benefit of incorporating information about contextual entities. In comparison, our method explores the impact of contextual entities more implicitly by blanking those entities out. Hence, by combining their insight with the MTB training method, we aim to build a RE model that generalizes better.

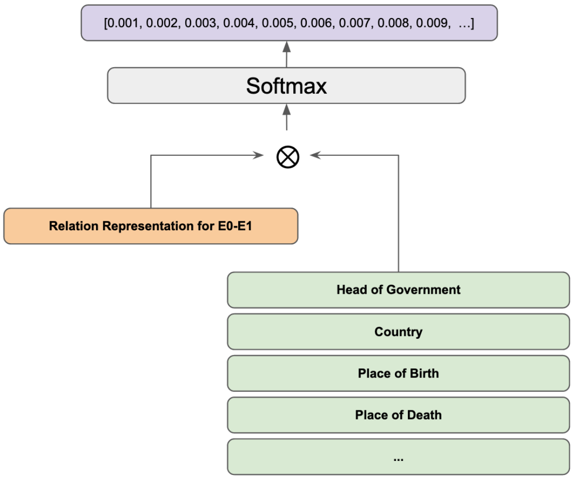

Figure 3: Per-class score

Similar to Soares et al., [9], Qin et al., 2021 [7] harness entity indicators to empower their relation extraction system. In addition to providing position information, their entity indicators include syntactic and semantic information as well. In this work, we stick to the entity marking scheme adopted by Soares et al., (2019) [9].

3 Architecture

To make the task more precise, we are given tokens $x_1, ..., x_N$ and coreferent entity spans $(s_i, f_i)_j$ where $j = 0, 1, 2$ ranges over entity indexes. The model then produces a probability distribution over relations. Figure 2 is an overview of the model architecture. The model architecture is adapted from the model presented in Soares et al. [9], with accommodations for multiple coreferent entity spans.

3.1 Preprocessing

For each entity index $j = 0, 1, 2$ and each span $(s_i, f_i)_j$, we wrap entity markers around the tokens in the span.

$$x_1 \; ... \; \texttt{[Ej]} \; x_{s_i} \; ... \; x_{f_i} \; \texttt{[/Ej]} \; ... \; x_N$$

`[Ej]` and `[/Ej]` are the entity start and end markers for entity j, respectively. If there are two

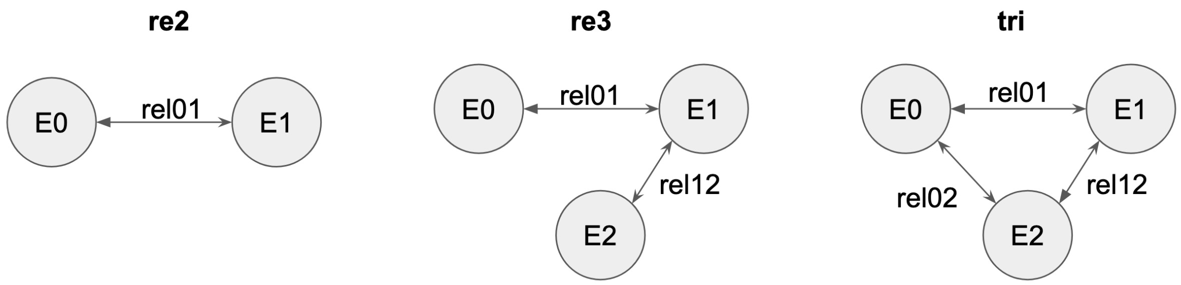

Figure 4: Depiction of the entities and edges involved in each training scheme.

or more entity spans for j, we wrap entity markers around both spans, adjusting indexes so each span refers to the correct tokens in the original token sequence. As done in Soares et al. [9], a hyperparamter $\alpha \in [0, 1]$ is fixed such that with probability α we randomly and independently choose whether to blank entity j, and if so replace every span of j with a `[BLANK]` token.

$$x_1 \dots \texttt{[Ej] [BLANK] [/Ej]} \dots x_N$$

With probability α^3 all three entities under consideration are blanked. The preprocessed input is then tokenized in the usual way for BERT language models, adding `[CLS]` and `[SEP]` tokens where necessary and splitting words into subtokens.

3.2 BERT with Entity Start Pooling

We then apply BERT (Devlin et al., 2018 [1]) to the tokenized input to produce hidden states $h_1, \dots, h_N$ for each token in the input sequence. For each entity index j, we take the hidden states for each occurrence of the entity start marker `[Ej]` and apply the component-wise max over each element to get the representation for that entity.

$$\vec{e}_j = \text{maxpool}\,\{\vec{h}_i \mid x_i = \texttt{[Ej]}\}$$

3.3 Per Class Score

To predict the relation between entities i and j, we concatenate entity representations $\vec{e}_i, \vec{e}_j$, then apply two linear transformations with bias to get the hidden relation representation R_{ij}.

$$R_{ij} = W_2(W_1(\vec{e}_i \oplus \vec{e}_j) + b_1) + b_2$$

To compute the per-class score, for each relation $k = 1, \dots, K$, the model memorizes a vector M_k of the same dimensions as R_{ij}. It then takes the dot product of R_{ij} with each relation vector M_k and normalizes the scores with softmax.

$$e_{ijk} = R_{ij} \cdot M_k$$

$$\vec{y}_{ij} = \text{softmax}\,[e_{ijk}]_{k=1}^{K}$$

where $[e_{ijk}]_{k=1}^{K}$ is a vector whose k-th element is e_{ijk}.

4 Experimental Setup

We train our model on three separate relation extraction tasks, which we call RE2, RE3, and TRI (Figure 4). RE2 is a multiclass setup wherein the model predicts a relation between two entities which are known to have some meaningful relation. In RE3, there are three entities and the model predicts the relation between two pairs of entities. In TRI, there are three entities and the model predicts the relation between all three pairs.

An important difference between the training methodology used by Soares et al., (2019) [9] and ours is that we only use the Matching the Blanks formulation as a fine-tuning step, whereas they use it to pre-train a BERT model in tandem with the masked language model objective.

We developed the model such that each of these training schemes could be applied independently across epochs. We trained several variations of the models with different values of α and different training schemes, each for three epochs. These variations are described in Table 2. Note that TRIBLANK[RE2×3] (Baseline) is trained with $\alpha = 0$ for 3 epochs and only on the RE2 task. This means `[BLANK]`, `[E2]`, and `[/E2]` are unseen tokens for this model. Similarly, TRIBLANK[RE2×3], $\alpha = 0.7$ has `[E2]` and `[/E2]` as unseen tokens. This will become more significant in the evaluation section.

Where possible, we chose hyperparameters to match Soares et al., 2019 [9].

- **Pretrained LM:** BERT-LARGE [1]
- **Optimizer:** ADAM [3]
- **Learning rate:** 2×10^{-6}

Scheme	# Train Examples	# Test Examples	# Train Entities	# Test Entities
RE2	38180	12323	28903	9496
RE3	39537	12345	18864	6073
TRI	15393	4476	8946	2804

Table 1: Dataset statistics for DOCRED schemes

Variation	α	First Epoch	Second Epoch	Third Epoch
TRIBLANK[RE2×3] (Baseline)	0	RE2	RE2	RE2
TRIBLANK[RE2×3]	0.7	RE2	RE2	RE2
TRIBLANK[RE3×3]	0.7	RE3	RE3	RE3
TRIBLANK[TRI×3]	0.7	TRI	TRI	TRI
TRIBLANK[RE2×2,RE3]	0.7	RE2	RE2	RE3
TRIBLANK[RE2,RE3,TRI]	0.7	RE2	RE3	TRI

Table 2: Variations of the TriBlank model. α determines the probability that an entity is blanked during training. Models are named according to how they are trained. For example, TRIBLANK[RE2×3] was trained for three epochs on RE2, and TRIBLANK[RE2×2,RE3] was trained for two epochs on RE2 and one epoch on RE3. Note that TRIBLANK[RE2×3] with $\alpha = 0$ is the BASELINE model.

- **Batch Size:** 3
- **Loss:** Cross Entropy
- **Framework:** PyTorch [6]
- **Hardware:** Google Colab Pro

We had to use a much smaller batch size due to memory limitations, which meant we had to decrease the learning rate as well. These two hyperparameters were fine-tuned by hand, though we would have preferred to conduct a more exhaustive grid-search.

5 Dataset

Since, as far as we can tell, the gold labels for the DOCRED [11] test dataset are not publicly available, we use the DOCRED train split for training and hyperparameter tuning, and the DOCRED validation split for testing. Table 1 contains statistics about our three training schemes extracted from DOCRED.

To create RE2, RE3, and TRI examples from the DOCRED dataset, we first choose two to three distinct entities [E0], [E1], and [E2], then identify any pairwise relations between them. It may initially seem odd that there are more RE3 training and test examples than RE2. However, consider that for a fully-connected directed graph with no self-edges, the number of paths of length 1 is $n(n-1) = O(n^2)$ whereas the number of paths of length 2 is $n(n-1)^2 = O(n^3)$. In the case of DOCRED, if entities are densely connected, there may be many more RE3 examples than RE2 examples. Heuristically, once [E0] and [E1] are chosen, there may be many possible choices for [E2].

In terms of dataset diversity, there are fewer entities represented in RE3 than RE2, and fewer still represented in TRI. Suppose [E0] has a relation to [E1], and neither [E0] nor [E1] is related to any other entity. Then [E0] and [E1] would appear in RE2, but not in RE3 or TRI. In fact, the entities in RE2 are a strict superset of the entities in RE3, which are a strict superset of the entities in TRI. As shown in Table 1, the number of unique entities in the dataset decreases from RE2 to RE3 to TRI.

6 Evaluation

We evaluated our model on a validation set from DocRED — both with and without blanking out entities in the validation text. Blanking out entities in the validation set means that the three entities under consideration are replaced by [BLANK] tokens, i.e. $\alpha = 1$ during evaluation.

6.1 Quantitative Analysis

We evaluated our model using the Macro F1 and Accuracy metrics.

6.1.1 Without entity blanking

Table 3 summarizes the results for the no-blanking evaluation setup. In the RE2 task, the BASELINE model TRIBLANK[RE2×3] with $\alpha = 0$

	Re2		Re3		Tri	
Variation	**Macro F1**	**Acc**	**Macro F1**	**Acc**	**Macro F1**	**Acc**
TriBlank[Re2×3] (Baseline)	**66.09**	**83.11**	64.21	82.24	**62.17**	81.61
TriBlank[Re2×3]	64.72	81.30	62.33	81.16	61.10	79.96
TriBlank[Re3×3]	60.27	79.74	62.85	82.42	58.72	81.25
TriBlank[Tri×3]	47.30	69.62	50.30	75.52	50.66	78.86
TriBlank[Re2×2,Re3]	64.79	82.51	**64.71**	**83.04**	59.94	81.86
TriBlank[Re2,Re3,Tri]	63.03	81.68	62.68	82.21	61.69	**82.48**

Table 3: Evaluation on the DocRED validation dataset using each scheme.

	Re2		Re3		Tri	
Variation	**Macro F1**	**Acc**	**Macro F1**	**Acc**	**Macro F1**	**Acc**
TriBlank[Re2×3] (Baseline)	**61.86**	**78.97**	60.18	77.48	56.95	75.71
TriBlank[Re2×3]	60.40	76.69	58.84	76.50	54.56	73.88
TriBlank[Re3×3]	56.95	75.74	59.77	78.62	54.98	77.17
TriBlank[Tri×3]	44.67	65.54	47.47	72.69	49.16	76.60
TriBlank[Re2×2,Re3]	60.08	78.16	**61.49**	**78.80**	55.59	77.52
TriBlank[Re2,Re3,Tri]	59.38	77.05	60.03	78.30	**57.41**	**78.39**

Table 4: Evaluation on the DocRED validation dataset using each scheme blanking all entities.

performs the best. This model was trained without any blanking. We see that in this task, models trained with blanking do worse. However, TriBlank[Re2×3] with $\alpha = 0.7$ and TriBlank[Re2×2, Re3] have the next best performances. This suggests that the Re2 training contributes the most to model performance in this evaluation setup. We expected this to be a consistent occurrence in further evaluations as well.

Interestingly, in the Re3 task, the best performing model was TriBlank[Re2×2, Re3]. This model does better than the expected TriBlank[Re3×3] model — indicating that the Re2 training is particularly helpful. In the Tri task, the TriBlank[Re2×3] achieves the highest Macro F1 score. (The TriBlank[Re2, Re3, Tri] indeed had the highest accuracy score but we consider Macro F1 as a more reliable metric due to class imbalances). These results indicate that not blanking out any contextual entities is indeed producing a better performing model. This supports our hypothesis that contextual entities do in fact impact the semantic relations between two target entities. Further, we observe that the TriBlank[Tri×3] model consistently does the worst across the three evaluation setups. This can be attributed to the difficulty of the training task (predicting three-way relations) and lack of sufficient training data.

In this evaluation setup, the models may have been able to rely on information from entity tokens and hence it becomes important to evaluate them using a blanking scheme in the validation setup as well.

6.1.2 With entity blanking

Table 4 summarizes the results for the with-blanking evaluation setup. For this setup, the models completely rely on context to predict relations between entities in each task. The target entities are blanked out. Similar to the previous setup, we have the TriBlank[Re2×3] ($\alpha = 0$) and TriBlank[Re2×2, Re3] models producing the best results on the Re2 and Re3 tasks respectively. It was surprising to see the BASELINE do the best for the with-blanking Re2 task because the model is trained without using any blanking. We believe that the Re2 training was indeed useful enough to leverage the context but maybe some hyper-parameter tuning on α might be able to improve other models.

There is roughly 3-5 point drop in both F1 and accuracy scores, indicating that the loss of the extra information from the entity tokens hurts model performance. In the Tri task, we find that the TriBlank[Re2, Re3, Tri] is the most performant, beating the TriBlank[Tri×3] model, which again has the worst results across the three tasks. We believe the

TRIBLANK[RE2, RE3, TRI] model leverages the pros of each training methodology well and hence in the adverse case of three blanked out entities is able to generalize well. Task specific training TRI in the last epoch acts almost like a fine-tuning for the model. For future work, we would like to further analyze these improvements by evaluating that model performance at different stages of training.

6.2 Qualitative Analysis

We analyze results from TRIBLANK[RE2,RE3,TRI] evaluated on the RE2 task without blanking. We randomly sample 100 examples where the model makes an incorrect prediction. The most cogent observation is that the predicted and gold labels are very often semantically related. For example, mistakes often involved confusion between the following sets of labels:

- country / located in the administrative territorial entity / country of citizenship / capital of
- author / creator
- producer / performer
- cast member / screenwriter

Of course, there were a few nonsensical predictions, though the majority of mistakes involved semantically related labels. Making these distinctions more clear is certainly an area of future work.

7 Future Work

1. *Predicting no relation between entities.* The TRIBLANK model assumes that all entities are related in some way. We can break this assumption by adding a ”None“ relation.

2. *Further training with more epochs.* The TRIBLANK[RE2,RE3,TRI] and TRIBLANK[RE2×2,RE3] models were able to outperform the TRIBLANK[RE2×3] model in certain areas. Perhaps training TRIBLANK for six epochs across RE2, RE3, and TRI could further boost performance.

3. *Break dependence on perfect coreference resolution.* The model assumes that coreferent entities are always tagged with entity markers, which may be an unrealistic assumption.

4. *Clearer distinctions between semantically related labels.* The relation types used in DOCRED have natural language descriptions. The `country` relation is described as

   > sovereign state of this item; don't use on humans

   [11] and the `located in the administrative territorial entity` relation is described as

   > the item is located on the territory of the following administrative entity. Use P276 (location) for specifying the location of non-administrative places and for items about events

   [11]. The TRIBLANK[RE2,RE3,TRI] often confused these two relation types. Perhaps by using these natural language descriptions and by embedding the labels in a space the model can perceive, we can increase performance.

8 Conclusion

In this work, we extended the Matching the Blanks training methodology introduced by Soares et al., 2019 [9] to three blanked out entities. This is based on the intuition that contextual entities impact the semantic relation between two target entities. Our results support the hypothesis that contextual entities impact semantic relations between target entities. However, we believe that more specialized models might be able to better harness that attribute. The code for our models and training setup is available here.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[2] Z. Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[3] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

[4] X. Li, F. Yin, Z. Sun, X. Li, A. Yuan, D. Chai, M. Zhou, and J. Li. Entity-relation extraction as multi-turn question answering, 2019.

[5] D. Lin and P. Pantel. Dirt: Discovery of inference rules from text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 323–328, New York, NY, 2001. ACM Press.

[6] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[7] Y. Qin, W. Yang, K. Wang, R. Huang, F. Tian, S. Ao, and Y. Chen. Entity relation extraction based on entity indicators. *Symmetry*, 13(4), 2021.

[8] G. Singh and P. Bhatia. Relation extraction using explicit context conditioning, 2019.

[9] L. B. Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski. Matching the blanks: Distributional similarity for relation learning, 2019.

[10] B. D. Trisedya, G. Weikum, J. Qi, and R. Zhang. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240, Florence, Italy, July 2019. Association for Computational Linguistics.

[11] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M. Sun. Docred: A large-scale document-level relation extraction dataset, 2019.