Exposing Shallow Heuristics of Relation Extraction Models with Challenge Data

Shachar Rosenman[1]   Alon Jacovi[1]   Yoav Goldberg[1,2]
[1]Computer Science Department, Bar Ilan University
[2]Allen Institute for Artificial Intelligence
{shacharosn, alonjacovi, yoav.goldberg}@gmail.com

Abstract

The process of collecting and annotating training data may introduce distribution artifacts which may limit the ability of models to learn correct generalization behavior. We identify failure modes of SOTA relation extraction (RE) models trained on TACRED, which we attribute to limitations in the data annotation process. We collect and annotate a challenge-set we call Challenging RE (CRE), based on naturally occurring corpus examples, to benchmark this behavior. Our experiments with four state-of-the-art RE models show that they have indeed adopted shallow heuristics that do not generalize to the challenge-set data. Further, we find that alternative question answering modeling performs significantly better than the SOTA models on the challenge-set, despite worse overall TACRED performance. By adding some of the challenge data as training examples, the performance of the model improves. Finally, we provide concrete suggestion on how to improve RE data collection to alleviate this behavior.

1 Introduction

In the relation extraction (RE) task, our goal is, given a set of sentences $s \in S$ to extract tuples (s, e_1, e_2, r) where a relation $r \in R$ holds between e_1 and e_2 (entities that appear in the sentence s, each represented as a span over s). RE is often represented as relation classification (RC): given a triplet of (s, e_1, e_2), determine which relation $r \in R$ holds between e_1 and e_2 in s, or indicate no-relation ($\emptyset$). This can be presented as a set of $|R|$ binary decision problems, $(s, e_1, e_2, r) \mapsto \{0, 1\}$: return 1 for tuples for which the relation holds, and 0 otherwise. The reduction between RE and RC is clear: given a sentence, extract all entity pair candidates (given a NER system), and run the RC problem on each of them.

Indeed, contemporary methods are all RC methods, and the popular TACRED large-scale relation extraction dataset is annotated for RC: each instance in the dataset is a triplet of (s, e_1, e_2) and is associated with a label $r \in R \cup \{\emptyset\}$. Importantly, the annotation is *non exhaustive*: not all e_1, e_2 pairs in the dataset are annotated (only 17.2% of the entity pairs whose type match a TACRED relation are). While this saves a lot of annotation effort, as we show this also leads to sub-optimal behavior of the trained models, and hinders our ability to properly assess their real-world utility.

We show that state of the art models trained on TACRED are often "right for the wrong reasons" (McCoy et al., 2019): instead of learning to perform the intended task, they rely on shallow heuristics which are effective for solving many dataset instances, but which may fail on more challenging examples. In particular, we show two concrete heuristics: classifying based on *entity types*, and classifying based on the *existence of an event without linking the event to its arguments*. We show that while they are not well attested in the dev and test sets, these challenging examples do occur in practice. We introduce CRE (Challenging RE), a challenge set for quantifying and demonstrating the problem, and show that four SOTA RC models significantly fail on the challenge set. We release the challenge set to encourage future research on better models. [1]

While we demonstrate the problem on TACRED, we stress that the model behaviors we expose are directly linked to the dataset construction procedure, and will likely occur in any dataset that is created in a similar fashion. We propose guidelines to help guide better datasets, and in particular better evaluation sets, in the future.

We also show that different modeling techniques

---

[1]GitHub repository with data and code: https://github.com/shacharosn/CRE

Heuristic	Examples	Prediction	Correct?
Event	(1) `Edward` was born in York in `1561` , the son of John, and his wife Mary.	*Birth date*	✓
	(2) Edward was born in York in `1561` , the son of `John` , and his wife Mary.	*Birth date*	✗
	(3) Edward was born in York in `1561` , the son of John, and his wife `Mary` .	*Birth date*	✗
Event	(4) `Loomis` is married to `Hilary Mills` , who wrote a biography about Norman Mailer.	*Spouse*	✓
	(5) `Loomis` is married to Hilary Mills, who wrote a biography about `Norman Mailer` .	*Spouse*	✗
	(6) `Loomis` is married to `Hilary Mills` , who wrote a biography about Norman Mailer.	*Spouse*	✓
	(7) `Loomis` is married to Hilary Mills , who wrote a biography about `Norman Mailer` .	*Spouse*	✗
	(8) Loomis is married to `Hilary Mills` , who wrote a biography about `Norman Mailer` .	*Spouse*	✗
	(9) Loomis is married to `Hilary Mills` , who wrote a biography about `Norman Mailer` .	*Spouse*	✗
Type & Event	(10) `UCF` also has `400` beds at the Rosen College Apartments Community, located on Rosen College of Hospitality Management campus.	*# Members*	✗
	(11) UCF also has `400` beds at the `Rosen College Apartments Community` , located on the Rosen College of Hospitality Management campus.	*# Members*	✗
	(12) UCF also has `400` beds at the Rosen College Apartments Community, located on the `Rosen College of Hospitality Management` campus.	*# Members*	✗

Figure 1: CRE dataset instances illustrating the various heuristics and error types. "# Members" refers to the number of human members or employees of an organization.

may alleviate this problem: models trained for QA are better at linking events to their arguments. While performing worse on TACRED overall, they perform significantly better on the challenge set.

2 Relation Classification Heuristics

McCoy et al. (2019) discusses the concept of "model heuristics"—decision rules that are used by ML models to score high on a test set, but which are too simplistic to solve the underlying problem—and demonstrated such heuristics used by NLI models. In this work we demonstrate model heuristics used by TACRED-trained RC models. Recall that a relation classification instance is $(s, e_1, e_2, r) \mapsto \{0, 1\}$.

***Event* Heuristic:** Classify based on (s, r). This heuristic ignores the entities altogether, acting as a classification model answering the question "does the sentence attest the relation". This heuristic is of limited applicability, as many sentences attest more than a single related pair of entities.

***Type* Heuristic:** Classify based on $(type(e_1), type(e_2), r)$, where $type(e)$ is the named-entity type of entity e. In a given dataset, a decision can be made based on the type of entities alone. For example of the 41 relations in the TACRED dataset, only the *per:religion* relation is between a *PERSON* and a *RELIGION*. A model may learn to incorrectly rely on the types when making a decision, ignoring the sentence s altogether.[2]

Many type-pairs are compatible with multiple relations in a dataset, weakening the utility of this heuristic. However, for applicable type-pairs, it can be very effective. For example, out of 21,284 Wikipedia sentences containing a PERSON name and a RELIGION name, 8,156 (38%) were classified by a RoBERTA-based RC model as *per:religion*. Manual inspection of a random sample of 100 of these, found that 42% are false-positives.

***Event+Type* Heuristic:** The *event* and *type* heuristics can be combined, by requiring the two decision rules $(s, r) \mapsto \{0, 1\}$ and $(type(e_1), type(e_2), r) \mapsto \{0, 1\}$ to hold. The resulting heuristic verifies that the sentence mentions the relation, and that the entity pairs of interest are type-compatible with the relation; it *does not verify* that the entities are arguments of the relation.

We demonstrate that the event+type heuristic is a particularly strong one for relation-classification datasets, and is widely used by trained state-of-the-art relation classifiers.

3 Challenge Set

Consider the *date_of_birth* relation, that holds between a person$_{e_1}$ and a year$_{e_2}$, and the classification instance:

[$_{e_1}$Steve Jobs] was born in California in [$_{e_2}$1955].

A model making use of the event+type heuristic will correctly classify this relation.[3] The heuristic is challenged by sentences that include multiple entities of an applicable type. For example:

[$_{e_1}$Ed] was born in [$_{e_2}$1561], the son of John, a carpenter, and his wife Mary.

A model relying solely on the event+type heuristic will correctly classify the above, but also incorrectly classify the following instances:

Ed was born in [$_{e_2}$1561], the son of [$_{e_1}$John], a carpenter, and his wife Mary.

---

[2]Another, related, heuristic is "classify based on (e_1, e_2, r)", that is, based on prior knowledge about the entities. For a discussion of related problem see (Shwartz et al., 2020). This is beyond the scope of the current work.

[3]Models that use the event heuristic on its own may predict *birth_location* instead, and models that rely on the type heuristic may predict *date_of_death*.

Ed was born in York in [$_{e_2}$1561], the son of John, a carpenter, and his wife [$_{e_2}$Mary].

While such sentences are frequent in practice, these cases are not represented in the dataset: in only 17.2% of the sentences in TACRED more than a single pair is annotated (and only 3.46% of the sentences have more than one different annotated labels in the sentence) . Additionally, due to the data collection method (Zhang et al., 2017), if a sentence includes a positive pair for a relation $r \in R$, it is significantly more likely that this pair will be chosen for annotation rather than a matching-type pair with a no-relation label. In other words, the data collection process leads to *no-relation* labels between a pair of entities being assigned with very high probability to sentences in which other pairs in the sentence also do not adhere to a relation of interest.

As a result, for models trained on TACRED, the model is incentivized to learn to identify the existence of the relation in the sentence, irrespective of the arguments. There is no signal in the data to incentivize the model to distinguish cases where the relation holds between the given arguments, from cases where the relation holds but between a different pair of arguments. We expect the same to hold for *any* large-scale RC dataset created in a similar manner.[4]

Challenge Set Construction. We seek a benchmark to highlight RC models' susceptibility to the *event+type* heuristic. The benchmark takes the form of a challenge/contrast set; a collection of related examples that specialize in a specific failure case, meant only for evaluation and not training (Kaushik et al., 2020; Gardner et al., 2020). In contrast to the NLI challenge set of McCoy et al. (2019), in which challenge instances are based on synthetic examples, our benchmark is based on naturally occurring examples from real-world corpora.

Methodology. Coming up with a set of real-world sentences that demonstrates a failure mode is not easy. The main challenge is in identifying potential candidate sentences to pass to manual annotation. To identify such cases, we require an effective method for sampling a population that is likely to exhibit the behavior we are interested in.

We propose the following challenge-set creation methodology: (1) use a strong *seed-model* to perform large-scale noisy annotation; (2) identify suspicious cases in the model's output; (3) manually verify (annotate) suspicious cases.

In stage (2), we identify suspicious cases by looking for sentences in which: (a) there are at least two entity-pairs of a NE type which is compatible to a TACRED relation (in most of the cases, the entity-pairs share one of the items, i.e. (e_1, e_2), (e_1, e_3)); and (b) these two pairs were assigned by the seed model to the same relation. Note that cases that satisfy condition (b) have, with high probability, at least one incorrect model prediction. On the other hand, given a strong seed model, there is also a high probability that one of the predictions is correct.

For our *seed-model* we use a SOTA relation classification model which is based on fine-tuned Span-BERT (Joshi et al., 2019), which we run over a large corpus of English Wikipedia sentences.[5]

Out of the sentences that passed stage (2), we randomly sampled 100 sentences for each of the 30 relation predicted by the model. All instance entity pairs were manually labeled by two of the authors of this work, while closely adhering to the TACRED annotation guideline. For each instance, the annotators provide a *binary* decision: does the entity pair in question adhere to relation r (as predicted by the model) or not.[6]

The resulting challenge set (CRE) has 3,000 distinct sentences, and 10,844 classification instances. The dataset is arranged into 30 groups of 100 sentences, where each sentence is binary labeled for a given relation. Example sentences from the challenge-set are given in Figure 1. In 57% of the sentences, there are at least two classification instances with conflicting labels, indicating the use of the event+type heuristic. On average there are 3.7 candidate entity-pairs per sentence. In 89.2% of the sentences in the set, the entity-pairs share an argument. Further details are available in the supplementary material.

---

[4]Annotation of entity pairs, if the pairs are chosen at random or all pairs are annotated, carries a significant inefficiency in the number of annotations of related pairs vs. number of annotations of unrelated pairs. For this reason, RE annotation setups such as TACRED (Zhang et al., 2017) implement bias towards labeling pairs that are likely to be related, and thus, a no-relation label implies a higher likelihood of no relation event occurring in the sentence.

[5]Additional details in the supplementary material.

[6]We chose to perform binary annotation, as we find it makes the annotation process faster and more accurate. As demonstrated by Alt et al. (2020), multi-class relation labeling by crowd-workers lead to frequent annotation errors. We observed the same phenomena also with non-crowd workers.

Model	P	R	F_1
RC-SpanBERT	70.8	70.9	70.8
RC-BERT	67.8	67.2	67.5
RC-KnowBERT	71.6	71.4	71.5
RC-RoBERTa	70.17	72.36	71.25

Table 1: Test results on TACRED.

4 Evaluating RE Models

By construction, the CRE dataset includes many instances that fail the seed RC model, which is based on fine-tuned SpanBERT (Joshi et al., 2019). To verify that the behavior is consistent across models, we evaluate also RC models fine-tuned over other state-of-the-art LMs: BERT (Devlin et al., 2019), RoBERTA (Liu et al., 2019b) and Know-BERT (Peters et al., 2019). When evaluated on TACRED test set (Table 1), these models achieve SOTA scores.

We evaluate model's results on the CRE dataset in terms of accuracy. We also report *positive accuracy (Acc_+)* (the accuracy on instances for which the relation hold; models that make use of the heuristic are expected to score high here) and likewise *negative accuracy (Acc_-)* (accuracy on instances in which the relation does not hold; models using the heuristic are expected to score low here). The models are consistently more accurate on the positive set then on the negative set showing that models struggle on cases where the heuristic make incorrect predictions.[7]

A direct comparison to the state-of-the-art is difficult with these metrics alone. To facilitate such a comparison, we also report precision, recall and F1 scores on **TACRED+Positive** (Table 3a), in which we add to the TACRED test set all the positive instances from the CRE dataset (easy cases), and on **TACRED+Negative** (Table 3b), in which we add the negative instances (hard cases). All models benefit significantly from the positive setting, and are hurt significantly in the negative setting *primarily in precision*, indicating that they do follow the heuristic on many instances:

the TACRED-trained models often classify

Model	Acc	Acc_+	Acc_-
RC-SpanBERT	63.5	89.7	42.5
RC-BERT	67.1	70.0	64.8
RC-KnowBERT	72.4	84.2	62.9
RC-RoBERTa	73.1	82.9	65.3
QA-SpanBERT	75.5	71.5	78.7
QA-BERT	67.4	62.9	70.9
QA-ALBERT	75.3	71.5	78.8

Table 2: CRE accuracy for the RE and QA models. Acc_+ refers to accuracy on positive instances. Acc_- refers to accuracy on negative instances.

Model	P	R	F_1
(a) TACRED + Positive			
RC-SpanBERT	88.2	79.3	83.5
RC-BERT	88.0	67.7	76.5
RC-KnowBERT	87.5	78.3	82.7
RC-RoBERTa	86.6	78.8	82.5
(b) TACRED + Negative			
RC-SpanBERT	43.3	70.9	53.8
RC-BERT	42.0	64.0	50.7
RC-KnowBERT	43.9	71.6	54.4
RC-RoBERTa	43.6	72.7	54.5

Table 3: P/R/F_1 scores on TACRED test set + all positive instances from the CRE dataset (a), and + all negative instances (b).

based on the type of arguments and the existence of a relation in the text, without verifying that the entities are indeed the arguments of the relation.

5 QA Models Perform Better

The CRE dataset results indicate that RE-trained models systematically fail to link the provided relation arguments to the relation mention. We demonstrate that QA-trained models perform better in this respect. The QA models differ in both in their training data (SQuAD 2.0, (Rajpurkar et al., 2018)) and in their training objective (span-prediction, rather than classification). Inspired by Levy et al. (2017) we reduce RC instances into QA instances. We follow the reduction from Cohen et al. (2020) between QA and binary relation classification which works by forming two questions for each relation instance, one for each argument. For example, for the relation instance pair *(Mark, FB, founded)* we ask "Who founded FB?" and "what did Mark found?".[8]

---

[7]CRE is binary labeled and relatively balanced between positive and negative examples, making accuracy a valid and natural metric. We chose to report Acc_+ and Acc_- instead of the popular precision and recall because precision and recall emphasize the positive class, and do not tell the full story of the negative class (indeed, prec/rec do not involve the true-negative case), which is of interest to us. Using the $Acc_{+/-}$ metric allows us to focus on both the positive and negative classes.

[8]The question templates for each relation are defined manually. The full set, and additional details on the reduction method, are given in Appendix D.

If the QA model answers either one of the questions with the correct span, we return 1 (relation holds), otherwise we return 0 (relation does not hold). We use three pre-trained SOTA models, fine-tuned on SQuAD 2.0: QA-SpanBERT, QA-BERT, and QA-AlBERT (Lan et al., 2020).

Results. While the scores on the TACRED test set are unsurprisingly substantially worse (with F_1 of 59.1%, 52.0% and 61.4%) than the TACRED trained models, they also perform better on the CRE dataset (Table 2).

QA-trained models pay more attention to the relation between an event and its arguments than RC-trained models.

6 Augmenting the Training Data with Challenge Set Examples

We test the extent by which we can "inoculate" (Liu et al., 2019a) the relation extraction models by enhancing their training data with some examples from the challenge-set. We re-train each model on the TACRED dataset, which we augment with half of the challenge-set (5504 examples, 8% of the size of the original TACRED training set). The other half of the challenge-set is used for evaluation.

Results. We begin by evaluating the inoculated models on the original TACRED evaluation set, establishing that this result in roughly the same scores, with small increases for most models (RC-SpanBERT: 71.0 F_1 (original: 70.8), RC-BERT: 69.9 F_1 (original:67.5), RC-KnowBERT: 72.1 F_1 (original:71.5), RC-RoBERTa: 70.8 F_1 (original:71.25)).

When evaluating on the CRE dataset examples, we see a large increase in performance for the inoculated dataset, as can be seen in Table 4. Compared to TACRED-only scores, While we see a small and expected drop for **Acc$_+$**, it is accompanied by a very large improvement on **Acc$_-$**. However, while accuracies improve, there is still a wide gap from perfect accuracy.

7 Discussion and Conclusion

We created a challenge dataset demonstrating the tendency of TACRED-trained models to classify using an event+type heuristic that fails to connect the relation and its arguments. QA-trained models are less susceptible to this behavior. Continuing Gardner et al. (2020), we conclude that challenge sets are an effective tool of benchmarking against

Model	Acc	Acc_+	Acc_-
(a) Trained on TACRED			
RC-SpanBERT	62.8	89.5	41.6
RC-BERT	65.8	68.4	63.7
RC-KnowBERT	71.6	83.0	62.5
RC-RoBERTa	75.5	85.4	68.0
(b) Trained on TACRED + half CRE			
RC-SpanBERT	84.4	85.7	83.4
RC-BERT	78.7	86.1	72.7
RC-KnowBERT	82.4	81.9	82.7
RC-RoBERTa	83.0	83.4	82.6

Table 4: $Acc/Acc_+/Acc_-$ scores on half of the CRE dataset, models trained on TACRED training set (a), models trained on TACRED training set with examples from the second half of the CRE dataset (b).

shallow heuristics, not only of models and systems, but also of data collection methodologies.

We suggest the following **recommendation for future RE data collection**: *evaluation sets* should be exhaustive, and contain *all* relevant entity pairs. Ideally, the same should apply also to *training sets*. If impractical, the data should at least attempt to exhaustively annotate *confusion-sets*: if a certain entity-pair is annotated in a sentence, all other pairs of the same entity-types in the sentence should also be annotated.

Acknowledgements

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. *CoRR*, abs/2004.14855.

Amir Cohen, DN, Shachar Rosenman, and Yoav Goldberg. 2020. Relation extraction as two-way span-prediction. *arXiv preprint*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Mohamed El-Geish. 2020. Gestalt: a stacking ensemble for squad2.0.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hanna Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating NLP models via contrast sets. *CoRR*, abs/2004.02709.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Divyansh Kaushik, Eduard H. Hovy, and Zachary Chase Lipton. 2020. Learning the difference that makes A difference with counterfactually-augmented data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019a. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2171–2179, Minneapolis, Minnesota. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, IV Logan, L Robert, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Vered Shwartz, Rachel Rudinger, and Oyvind Tafjord. 2020. " you are grounded!": Latent name artifacts in pre-trained language models. *arXiv preprint arXiv:2004.03012*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Cuihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

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*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

A   Challenge Set

We use SpanBERT as a seed model, SpanBert is a recent state-of-the-art model uses pre-trained language model fine-tuned to the RC task which uses a bidirectional LM pre-trained on SpanBERT We run the seed model over a large corpus of English Wikipedia sentences in which contains more than 10 million sentences. Table 8 contain all the relations in the challenge set with number of positive and negative instances per relation. The challenge set is relatively balanced in terms of positive examples (44%) and negative examples (56%), and the classification is binary, so it is possible to use accuracy as evaluation method. The average length of a sentence is 28.7 tokens. Crucially, there are no sentences with less than two pair of entities.

B   Experiment Details

RC models   We evaluate four RC models fine-tuned on TACRED: **SpanBERT** (Joshi et al., 2019) extends BERT model by pre-trained on a span-level; **KnowBERT** (Peters et al., 2019) which integrates knowledge bases into BERT, we use KnowBERT-W+W, where the knowledge comes from joint entity linking and language modelling on Wikipedia and WordNet; **RoBERTa** (Liu et al., 2019b), we use the same baseline from (Wang et al., 2020); **BERT** (Devlin et al., 2019) we use the same model from (Joshi et al., 2019) for BERT base. Table 1 contain the performance of the four models on TACRED.

QA models   We use three pre-trained SOTA models, fine-tuned on SQuAD 2.0: **QA-SpanBERT**: SpanBERT (Joshi et al., 2019) we use a pre-trained model (Joshi et al., 2019) fine-tuned on SQuAD 2.0 for QA (F_1: 88.7). **QA-BERT**: BERT (Devlin et al., 2019), we use a pre-trained model (Joshi et al., 2019) fine-tuned on SQuAD 2.0 for QA. (F_1:

83.3) **QA-ALBERT**: ALBERT xxlarge (Lan et al., 2020), we use a pre-trained model (El-Geish, 2020) fine-tuned on SQuAD 2.0 for QA. (F_1: 88.9)

C   How we evaluate TACRED + pos/neg

In order to evaluate TACRED + Positive and TACRED+Negative we presented them as binary decision problem as explained in section 1. We simulate ACRED + pos/neg in which for each relation $r \in R$ we create a set that contain all the examples $(s, e_1, e_2, r) \mapsto \{0, 1\}$, which the arguments types e_1 and e_2 match the relation r. We evaluate each set of relation r separately, and we report the results as micro-averaged F1 scores.

D   Detailed QA reduction

Table 7 contain the templates for the questions, each relation have two questions, a question for the head entity and a question for the tail entity, we use this templates to reduce the relation classification task.

E   Detailed Results

Table 5, 6 shows the full results of all the models on the challenge set, and table 6 shows the full results of all the models on **TACRED+Positive** and **TACRED+Negative**, Δ denotes the difference in performance between them, All models preformed a big difference in precision between the positive and negative examples.

Model	P	R	F_1	TP	FP	TN	FN	Pos_acc	Neg_acc	Total_acc
RC-SpanBERT	55.6	89.7	68.6	39.9	31.8	23.6	4.5	89.7	42.5	63.5
RC-BERT	61.0	70.0	65.5	31.1	19.5	36.0	13.3	70.0	64.8	67.1
RC-KnowBERT	64.5	84.1	73.0	37.4	20.5	34.9	7.0	84.2	62.9	72.4
RC-RoBERTa	65.7	82.8	73.3	36.8	19.2	36.3	7.6	82.9	65.3	73.1
QA-SpanBERT	72.9	71.5	72.2	31.8	11.7	43.7	12.6	71.5	78.7	75.5
QA-BERT	63.4	62.9	63.0	28.0	16.1	39.4	16.4	62.9	70.9	67.4
QA-ALBERT	72.6	71.5	72.0	31.8	12.0	43.5	12.6	71.5	78.3	75.3

Table 5: Results of models on the CRE dataset.

Models	+ positive			+ negative			Difference		
	P	R	F_1	P	R	F_1	Δ_P	Δ_R	Δ_{F_1}
RC-SpanBERT	88.2	79.3	83.5	43.3	70.9	53.8	44.9	8.4	29.7
RC-BERT	88.0	67.7	76.5	42.0	64.0	50.7	46.0	3.7	25.8
RC-KnowBERT	87.5	78.3	82.7	43.9	71.6	54.4	43.6	6.7	28.3
RC-RoBERTa	86.6	78.8	82.5	43.6	72.7	54.5	43.0	6.1	28.0
QA-SpanBERT	70.6	67.1	68.8	36.0	60.7	45.2	34.6	6.4	23.6
QA-BERT	66.3	63.7	65.0	32.7	64.9	43.4	33.6	-1.2	21.6
QA-ALBERT	80.5	70.0	74.9	41.9	63.0	50.3	38.6	7.0	24.6

Table 6: RE and QA models on TACRED + negative, TACRED + positive, and the differences between them.

Relation	Question 1	Question 2
org:founded_by	Who founded e_1?	What did e_2 found?
per:employee_of	Where does e_1 work?	Who is an employee of e_2?
per:title	What is e_1's title?	Who has the title e_2?
per:age	What is e_1's age?	Whose age is e_2?
per:date_of_birth	When was e_1 born?	What is e_2's date of birth?
org:top_members/employees	Who are the top members of the organization e_1?	e_2 is a top member of which organization?
org:country_of_headquarters	in what country the headquarters of e_1 is?	What is the country that e_2's headquarters located in?
per:parents	Who are the parents of e_1?	Who are the parents of e_2?
per:countries_of_residence	What country does e_2 resides in?	In what country does e_2 live
per:children	Who are the children of e_1?	Who are e_2's children?
org:alternate_names	What is the alternative name of the organization e_2?	What are other names for e_2?
per:charges	What are the charges of e_1?	What is e_2 charged with?
per:cities_of_residence	What city does e_2 resides in?	What is e_2's cities of residence?
per:origin	What is e_1 origin?	Where's e_2's origin?
per:siblings	Who is the sibling of e_1?	Who is the brother of ?
per:alternate_names	What is the alternative name of e_1?	is another name for which person?
org:website	What is the URL of e_1	what is the website address of ?
per:religion	What is the religion of e_1?	What religion does e_2 believe in?
per:stateorprovince_of_death	Where did e_1 died?	What is the place where e_2 died?
org:parents	What organization is the parent organization of e_1?	What organization is the parent organization of e_2?
org:subsidiaries	What organization is the child organization of e_1?	What organization is the child subsidiaries of e_2?
per:other_family	Who are family of e_1?	ho are family of e_2?
per:stateorprovinces_of_residence	What is the state of residence of e_1?	Where is e_2's place of residence?
org:members	Who is a member of the organization e_1?	What organization e_2 is member of?
per:cause_of_death	How did e_1 died?	What is e_2's cause of death?
org:member_of	What is the group the organization e_1 is member of?	What organization is a member of e_2?
org:number_of_employees/members	How many members does e_1 have?	What is the number of members of e_2?
per:country_of_birth	In what country was e_1 born	hat is e_2's country of birth?
org:shareholders	Who hold shares of e_1?	Who are e_2's shareholders?
org:stateorprovince_of_headquarters	What is the state or province of the headquarters of e_1?	Where is the state or province of the headquarters of e_2?
per:city_of_death	In what city did e_1 died?	What is e_2's city of death?
per:city_of_birth	In what city was e_1 born?	What is e_2's city of birh?
per:spouse	Who is the spouse of e_1?	Who is the spouse of e_2?
org:city_of_headquarters	Where are the headquarters of e_1?	What is e_2's city of headquarters?
per:date_of_death	When did e_1 die?	What is e_2's date of death?
per:schools_attended	Which schools did attend?	What school did e_2 attend?
org:political/religious_affiliation	What is e_1 political or religious affiliation?	What religion does e_1 organization belong to?
per:country_of_death	Where did e_1 die?	What e_2's country of death?
org:founded	When was e_1 founded?	What date did e_2 establish?
per:stateorprovince_of_birth	In what state was e_1 born?	What is e_2's country of birth?
per:city_of_birth	Where was e_1 born?	Who was born in e_2?
org:dissolved	When was e_1 dissolved?	What date did e_2 dissolved?

Table 7: Templates for the questions, for each relation two questions are defined.

Relation	Positive	Negative
org:founded_by	214	179
per:age	114	186
per:date_of_birth	150	158
org:founded	123	190
per:schools_attended	193	175
per:employee_of	209	120
org:country_of_headquarters	136	120
per:alternate_names	390	210
per:children	141	459
per:city_of_birth	108	148
per:city_of_death	119	103
per:date_of_death	112	153
per:religion	174	77
per:spouse	212	388
org:alternate_names	357	243
per:title	126	111
org:parents	141	459
per:other_family	332	268
per:stateorprovince_of_birth	121	105
org:political/religious_affiliation	119	101
per:siblings	232	368
per:origin	153	129
per:cities_of_residence	102	157
org:city_of_headquarters	129	121
per:countries_of_residence	95	152
per:parents	142	458
per:stateorprovinces_of_residence	110	154
org:top_members/employees	89	189
org:stateorprovince_of_headquarters	135	86
org:number_of_employees/members	41	258
Total	4819	6025

Table 8: CRE dataset number of positive and negative instances per relation