REBEL: Relation Extraction By End-to-end Language generation

Pere-Lluís Huguet Cabot
Sapienza University of Rome
& Babelscape, Italy
huguetcabot@babelscape.com

Roberto Navigli
Sapienza University of Rome
navigli@diag.uniroma1.it

Abstract

Extracting relation triplets from raw text is a crucial task in Information Extraction, enabling multiple applications such as populating or validating knowledge bases, factchecking, and other downstream tasks. However, it usually involves multiple-step pipelines that propagate errors or are limited to a small number of relation types. To overcome these issues, we propose the use of autoregressive seq2seq models. Such models have previously been shown to perform well not only in language generation, but also in NLU tasks such as Entity Linking, thanks to their framing as seq2seq tasks. In this paper, we show how Relation Extraction can be simplified by expressing triplets as a sequence of text and we present REBEL, a seq2seq model based on BART that performs end-to-end relation extraction for more than 200 different relation types. We show our model's flexibility by fine-tuning it on an array of Relation Extraction and Relation Classification benchmarks, with it attaining state-of-the-art performance in most of them.

1 Introduction

Extracting relational facts from text has been an ongoing part of Natural Language Processing. The ability to extract semantic relationships between entities from text can be used to go from unstructured raw text to structured data that can be leveraged in an array of downstream tasks and applications, such as the construction of Knowledge Bases.

Traditionally this task has been approached as a two-step problem. First, the entities are extracted from text as in Named Entity Recognition (NER). Second, Relation Classification (RC) checks whether there exists any pairwise relation between the extracted entities (Zeng et al., 2014; Zhang et al., 2017). However, identifying which entities truly share a relation can become a bottleneck, requiring additional steps such as negative sampling and expensive annotation procedures.

More recently, end-to-end approaches have been used to tackle both tasks simultaneously (Miwa and Sasaki, 2014; Pawar et al., 2017; Katiyar and Cardie, 2017; Eberts and Ulges, 2020). This task is usually referred to as Relation Extraction or End-to-End Relation Extraction (RE). In this scenario, a model is trained simultaneously on both objectives. Specific parts of the model can be assigned different tasks of the pipeline, such as NER, on the one hand, and classifying the relations between the predicted entities (RC), on the other. By training both tasks simultaneously, the model benefits from the information bias between the tasks as in multi-task setups (Caruana, 1998), improving performance on the end-to-end RE task.

Although successful, these models are often complex, with task-focused elements that need to be adapted to the number of relation or entity types, or they are not flexible enough to work for texts of different nature (sentence vs. document level) or domains. Moreover, they usually require long training times in order to be fine-tuned on new data.

In this paper, we present REBEL (Relation Extraction By End-to-end Language generation), an autoregressive approach that frames Relation Extraction as a seq2seq task, together with the REBEL dataset, a large-scale distantly supervised dataset, obtained by leveraging a Natural Language Inference model. Our approach provides some upsides over previous end-to-end approaches thanks to our adoption of a simple triplet decomposition into a text sequence. By pre-training an Encoder-Decoder Transformer (BART) using our new dataset, REBEL achieves state-of-the-art performance on an array of RE baselines within a few epochs of fine-tuning. Its simplicity makes it highly flexible to adapt to new domains or longer documents. As the same model weights are still utilized after the pre-training phase, there is no need to train model-specific components from scratch, making training more efficient.

Moreover, although it is devised for Relation Extraction, the same approach can be generalized to Relation Classification, achieving competitive results.

We make REBEL available[1] both as a standalone model that can extract more than 200 different relation types, and as a pre-trained RE model that can be easily fine-tuned on new RE and RC datasets. We also provide the REBEL dataset and the pipeline to extract high-quality RE datasets from any Wikipedia dump.

2 Related work

2.1 Relation Extraction

The term Relation Extraction is often used in the literature for different tasks and setups in the literature (Taillé et al., 2020). For clarity, we refer to Relation Extraction (RE) as the task of extracting triplets of relations between entities from raw text, with no given entity spans, usually also called end-to-end Relation Extraction. We refer to classifying the relation between two entities in a given context as Relation Classification (RC).

Early approaches tackled RE as a pipeline system, identifying the entities present in the text using Named Entity Recognition, and then classifying the relation, or lack of, between each pair of entities present in the text (RC). Therefore, early work made use of CNNs or LSTMs to exploit sentence-level semantics and classify the relations between two given entities (Zeng et al., 2014; Zhou et al., 2016). Current approaches to Relation Classification use Transformer models, with (Yamada et al., 2020) being the current state of the art by enhancing BERT (Devlin et al., 2019) with entity-aware components.

Early end-to-end approaches using neural networks classified all word pairs present in the input text (Miwa and Sasaki, 2014; Pawar et al., 2017) using table representation, or table filling, re-framing the task into filling the slots of a table (the relations) where rows and columns are the words in the input. More recently, Wang and Lu (2020) used a similar table-based formulation, where the table is explicitly encoded using a table-sequence encoder.

Finally, there are pipeline systems that tackle both parts of Relation Extraction, NER, and RC, by jointly training components that take advantage of the information shared between the tasks. In these setups, entities are first extracted as in NER using BILOU tags and then a biaffine classifier extracts their relations, sharing part of the encoders for both tasks. These range from LSTMs (Miwa and Bansal, 2016; Katiyar and Cardie, 2017) to CNNs (Adel and Schütze, 2017; Zheng et al., 2017) and, lately, Transformer-based architectures (Eberts and Ulges, 2020), that explicitly predict and encode entity spans instead of the BILOU approach used in NER.

All recent sentence-level RE models are based on Transformer models, such as BERT (Eberts and Ulges, 2020; Wang et al., 2020) or ALBERT (Lan et al., 2020; Wang and Lu, 2020). To tackle document-level RE, Eberts and Ulges (2021) use a pipeline approach jointly trained on a multi-task setup that leverages coreference resolution to operate at an entity level, rather than mentions.

While the aforementioned work highlights the relevance of Relation Extraction as a task, the lack of consistent baselines or a cohesive task definition has led to discrepancies in the use of datasets and the way models have been evaluated. Taillé et al. (2020) explain the different issues in-so-far, and also make an attempt to unify RE evaluation and perform a fair comparison between systems.

We will follow their guidelines and use strict evaluation, unless specified, for which a relation is considered correct only if the head and tail entity surface forms are correctly extracted (i.e., fully overlap with the annotation), as well as the relation and entity types (if available for the dataset).

2.2 Seq2seq and Relation Extraction

The pipeline and table filling methods described so far have proved to perform well on RE, but still face some challenges. They often assume at most one relation type between each entity pair, and multi-class approaches do not take other predictions into account. For instance, they could predict two "birth dates" for the same head entity, or predict relations that are incompatible together. Moreover, they require all possible entity pairs to be inferred, which can become computationally expensive.

Seq2seq approaches for RE (Zeng et al., 2018, 2020; Nayak and Ng, 2020) offer some off-the-shelf solutions to these problems. Decoding mechanisms can output the same entities multiple times, as well as conditioning future decoding on previous predictions, implicitly dealing with incompatible ones. However, as Zhang et al. (2020) discuss, they still pose some issues. The triplets need to

[1] https://github.com/babelscape/rebel

be linearized into a somewhat arbitrary sequential order, such as the alphabetical one. This issue is explored by Zeng et al. (2019), who use Reinforcement Learning to compute the extraction order for the triplets. Moreover, seq2seq approaches suffer from exposure bias, since at training time the prediction is always dependent on the gold-standard output. In Zhang et al. (2020) a tree-decoding approach mitigates these issues while still using an autoregressive seq2seq approach.

In the meantime, seq2seq Transformer models, such as BART (Lewis et al., 2020) or T5 (Raffel et al., 2020) have been used in NLU tasks such as Entity Linking (Cao et al., 2021), AMR parsing (Bevilacqua et al., 2021), Semantic Role Labeling (Blloshmi et al., 2021) or Word Sense-Disambiguation (Bevilacqua et al., 2020) by reframing them as seq2seq tasks. Not only do they show strong performance, but they also showcase the flexibility of seq2seq models by not relying on predefined entity sets, but rather on the decoding mechanism, which can easily be extended to new or unseen entities.

For our model, we employ an Encoder-Decoder framework that can alleviate some of the previous issues seq2seq for RE has faced. While exposure bias can still occur, the attention mechanism enables long-distance dependencies as well as attending (or not) to the previously decoded output. Additionally, we devise a novel triplet linearization with a consistent triplet ordering that enables the model to leverage both the encoded input and the already decoded output.

3 REBEL

We tackle Relation Extraction and Classification as a generation task: we use an autoregressive model that outputs each triplet present in the input text. To this end, we employ BART-large (Lewis et al., 2020) as the base model.

In a translation task, teacher forcing leverages pairs of text in two languages by conditioning the decoded text on the input. At training time the encoder receives the text in one language, and the decoder receives the text in the other language, outputting the prediction for the next token at each position.

In our approach, we *translate* a raw input sentence containing entities, together with implicit relations between them, into a set of triplets that explicitly refer to those relations. Therefore, we need to express the triplets as a sequence of tokens to be decoded by the model. We design a reversible linearization using special tokens that enable the model to output the relations in the text in the form of triplets while minimizing the number of tokens that need to be decoded.

For REBEL, we have as input the text from the dataset and, as output, the linearized triplets. If x is our input sentence and y the result of linearizing the relations in x as explained in Section 3.1, the task for REBEL is to autoregressively generate y given x:

$$p_{BART}(y \mid x) = \prod_{i=1}^{len(y)} p_{BART}\left(y_i \mid y_{<i}, x\right)$$

By fine-tuning BART on such a task, using the Cross-Entropy loss as in Summarization or Machine Translation, we maximize the log-likelihood of generating the linearized triplets given the input text.

3.1 Triplets linearization

For RE, we want to express triplets as a sequence of tokens such that we can retrieve the original relations and minimize the number of tokens to be generated so as to make decoding more efficient. We introduce a set of new tokens, as markers, to achieve the aforementioned linearization. `<triplet>` marks the start of a new triplet with a new head entity, followed by the surface form of that entity in the input text. `<subj>` marks the end of the head entity and the start of the tail entity surface form. `<obj>` marks the end of the tail entity and the start of the relation between the head and tail entity, in its surface form. To obtain a consistent order in the decoded triplets, we sort the entities by their order of appearance in the input text and linearize the triplets following that order. Triplets will also be grouped by head entity. Therefore, the first triplet will be the one with the first appearing head entity and the following relation will be the one with the first appearing tail entity related to that head entity, followed by the rest of triplets with the same head entity. There is no need to specify the head entity each time, reducing the decoded text length. Once there are no more relations with that head entity, a new group of relations will start, with the second appearing head entity in the text, repeating the same process until there are no more triplets to be linearized. This mechanism is described in Algorithm 1.

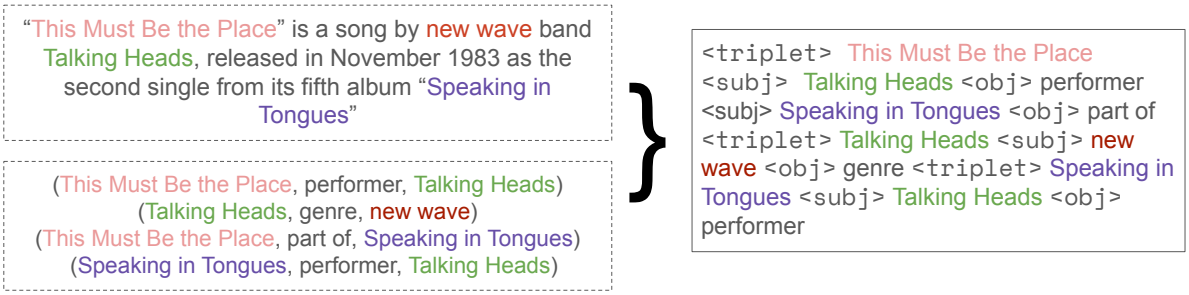

Figure 1: Example of the triplet linearization process for REBEL.

Algorithm 1: Transform a set of relations R into a text sequence

Result:
$lin\_triplets$ with all triplets as a sequence of text.
Input:
E = Entities;
R = Relations;
$sort()$ Sorts by placement in input text;
Start:
$E = sort(E)$;
$lin\_triplets$ = "";
for $e \in E$ **do**
  $R(e)$ = relations with e as subject;
  $R(e) = sort(R(e))$;
  $lin\_triplets$ += <triplet> + e;
  for $r \in R(e)$ **do**
    $o = E(e, r)$ object of relation r;
    $lin\_triplets$ += <subj> + o + <obj> + r;
  end
end

Figure 1 shows an example of the linearization process for a list of relations and an input sentence. Notice how *This Must Be the Place* appears twice as a subject, but it is present only once in the output as a subject entity. The original triplets can easily be retrieved by taking the special tokens into account. In RE datasets, the entity types are also present in the triplets and need to be predicted by the model. In that case, we apply a modification of Algorithm 1 where instead of <subj> and <obj>, we add new tokens for each entity type, such as <per> or <org>, for person or organization, respectively, and use them in the same fashion, indicating the type of the entity they follow.

3.2 REBEL dataset

Autoregressive transformer models such as BART or T5, have been shown to perform well on different generative tasks such as translation or summarization, but they do require large amounts of data to be trained. On the other hand, end-to-end relation extraction datasets are scarce and often small.

In Elsahar et al. (2018) the T-REx dataset was created by devising a pipeline that extracts entities and relations from DBpedia abstracts to overcome this lack of big RE datasets. While the result is a large dataset, the quality of the annotation presents some issues. First, the use of a somewhat old entity linking tool (Daiber et al., 2013) leads to entities being wrongly disambiguated. Since the relations are extracted by using those entities, this leads to missing or faulty relations. Moreover, most of the relations are extracted by assuming that, if the two entities are present in the text, the relation is therefore entailed by this presence.

We overcome these issues by expanding upon their pipeline to create a large silver dataset, used as pre-training for REBEL. We use Wikipedia[2] abstracts, that is, the part of each Wikipedia page before the table of contents, extracted using `wikiextractor` (Attardi, 2015). Then, we link the entities present in the text as hyperlinks, together with dates and values, to Wikidata entities using `wikimapper`[3]. From this, we extract all the relations present between those entities in Wikidata. Our system can be used with any Wikipedia dump, in multiple languages, enabling light and quick extraction using a multi-core process and SQL to avoid memory issues with the Wikidata dump.

However, a relation in Wikidata does not nec-

[2]Downloaded on 2021/02/01 from: https://dumps.wikimedia.org/enwiki/
[3]https://pypi.org/project/wikimapper/

	Entity Types	Relation Types	Train		Validation		Test	
CONLL04	4	5	1,290	(922)	343	(231)	422	(288)
NYT	3	24	94,222	(56,196)	8,489	(5,000)	8,616	(5,000)
DocRED	6	96	3,7486	(3,008)	3,678	(300)	8,787	(700)
ADE	2	1	6,821	(4,272)	-		-	
Re-TACRED	17	40	58,465	(58,465)	19,584	(19,584)	13,418	(13,418)
REBEL (sent.)	-	220	878,555	(784,202)	48,514	(43,341)	48,852	(43,506)
REBEL (full)	-	1,146	9,282,837	(2,754,387)	513,270	(152,672)	515,186	(152,835)

Table 1: Dataset statistics. Number of triplets with number of instances in parenthesis.

essarily mean that the relation is entailed within the text. Although in Elsahar et al. (2018) high reliability is claimed using this method, it has been shown to be noisy for frequent relations such as country or spouse, and we have found several related annotation issues. We utilize a pre-trained RoBERTa (Liu et al., 2019) Natural Language Inference (NLI) model[4] to tackle this issue, and use its entailment prediction to filter those relations not entailed by the Wikipedia text. For each triplet, we input the text containing both entities from the Wikipedia abstract, and the triplet in their surface forms, `subject` + `relation` + `object`, separated by the `<sep>` token.

For the previous example and the triplet (Talking Heads, genre, new wave), we input: *"This Must Be the Place" is a song by new wave band Talking Heads, released in November 1983 as the second single from its fifth album "Speaking in Tongues". <sep> Talking Heads genre new wave*. We keep those triplets for which the entailment prediction is higher than 0.75. This proves successful in creating cleaner data in preliminary experiments and removing noisy annotations. We create three random splits, with validation and test each being 5% of the total data.

While this data extraction pipeline may still keep some noise, or exclude some relations that are entailed by the text, it enables an automatic way of gathering millions of entities and relations as a silver dataset, sufficient for training our model. We name our RE dataset creation tool cRocoDiLe: Automatic Relation Extraction Dataset with NLI filtering, and we make it available here[5].

4 Experimental Setup

In this section, we describe the setup to train and evaluate REBEL for four different widely used RE datasets and one RC dataset. Statistics for all the datasets, including our pre-training dataset, can be found in Table 1.

While the training objective is on the autoregressive task, we evaluate the model on RE, extracting all the triplets from the generated output, and evaluating using Recall, Precision, and micro-F1 based on the labeled triplets. For a triplet to be considered correct, the entities and the relation, as well as their types, have to be the same as the labeled ones (this is known as "strict" evaluation in RE) using the evaluation code from Taillé et al. (2020).

4.1 REBEL dataset

We create this dataset by matching Wikipedia hyperlinks with Wikidata entities as explained in Section 3.2. To pre-train our model, we use a sentence-level version of it, where only relations between entities present in each sentence are kept. We keep the 220 most frequent relations in the train split.

We fine-tune REBEL (using BART-large as the base model) on the silver dataset for 6 epochs. We refer to the resulting model as REBEL$_{pre-training}$. While REBEL$_{pre-training}$ is in and of itself capable of extracting relations subsuming about 220 types, we show that it also functions as a base step for downstream RE and RC tasks, which are fine-tuned on top of it.

4.2 CONLL04

CONLL04 (Roth and Yih, 2004) is composed of sentences from news articles, annotated with four entity types (*person*, *organization*, *location* and *other*) and five relation types (*kill*, *work for*, *organization based in*, *live in* and *located in*). To compare with previous work, we use the test split from Gupta et al. (2016), and the same validation set as Eberts and Ulges (2020), although we do not include the validation set at final training time.

For CONLL04 we expand REBEL to include entity types. As described in Section 3.1, we introduce a set of new tokens for each entity type. For CONLL04 these are `<peop>`, `<org>`, `<loc>`,

[4] xlm-roberta-large-xnli

[5] https://github.com/Babelscape/crocodile

<other>. We fine-tune on top of REBEL for 30 epochs and test on the best performing epoch on the validation set.

4.3 DocRED

DocRED (Yao et al., 2019) is a recent dataset created similarly to our pre-training data, by leveraging Wikipedia and Wikidata. However, it focuses on longer spans of text, with relations between entities at a document level. There is a distantly supervised portion, while the validation and (hidden) test sets are manually annotated. It includes annotations for 6 different entity types and 96 relation types.

Despite the fact that DocRED was originally designed as a relation classification task, we use the splits from Eberts and Ulges (2021) and tackle it as a relation extraction task. In DocRED there are 6 entity types, consequently we use the tokens: <loc>, <misc>, <per>, <num>, <time> and <org> to indicate them.

We fine-tune on top of REBEL for 21 epochs and test on the last checkpoint, using a beam search of 10. For REBEL$_{pre-training}$, we use a version trained on a filtered dataset not including any of the Wikipedia pages present in DocRED validation or test sets.

4.4 NYT

NYT (Riedel et al., 2010) is a dataset consisting of news sentences from the New York Times corpus. The dataset contains distantly annotated relations using FreeBase. We use the processed version of Zeng et al. (2018) called NYT-multi, which contains overlapping entities, with three different entity types, and 24 relation types.

We use <loc>, <per> and <org> to indicate the 3 entity types. As for the 24 relation types, we map these to natural language expressions to match those seen at pre-training.

We fine-tune on top of REBEL for a maximum of 42 epochs and test on the best performing epoch on the validation set.

4.5 ADE

ADE (Gurulingappa et al., 2012) is a dataset on the biomedical domain, for which Adverse-Effects from drugs are annotated as pairs of drug and adverse-effect. The dataset provides 10-folds of train and test splits.

Drug and Adverse-Effect are the two entity types, and are always the subject and object entities for the single relation Adverse-Effect. Thus, we keep the same setup as with REBEL, using the <subj> token to distinguish between entity types, and removing the relation from the output, as it is always the same.

We fine-tune on top of REBEL for 25 epochs and evaluate using the last checkpoint for each fold in the dataset. Hyperparameters are selected by using 10% of the training data in the first fold.

4.6 Re-TACRED

Re-TACRED (Stoica et al., 2021) is a Relation Classification dataset, a revised version of the widely used TACRED (Zhang et al., 2017), fixing some of the issues pointed out by Alt et al. (2020). We want to extract the relation between two given entities, or the no_relation prediction, accounting for 63% of the 91,467 sentences in the dataset. To this end, we follow the approach from Zhou and Chen (2021) and Zhou and Chen (2021) and mark the entities in the input text using punctuation marks. We do not include any entity-type information.

The output is treated as in previous tasks, and we do not force the decoding of the given entities, as we find it is sufficient to mark them in the input. We fine-tune on top of REBEL for 8 epochs and evaluate using the last checkpoint.

5 Results

5.1 Relation Extraction

For our pre-training task using the REBEL dataset, the model achieves 74 micro-F1 and 51 macro-F1. The dataset is created by distant supervision and serves as a pre-training step, however, it is worth noting its performance for predicting up to 220 different relation types.

Results on selected baselines are presented in Table 2, as well as additional metrics in Tables 3 and 4. We see an improvement across all datasets with pre-trained REBEL, achieving between 1.2 and 6.7 absolute F1 points improvement over recent state-of-the-art models. Using REBEL without the pre-training, we see that performance decreases, especially for smaller datasets or those with many entity types. Nevertheless, it still achieves competitive results, showing the flexibility of tackling RE as a seq2seq task using Transformer Encoder-Decoder models.

Additionally, REBEL shows a better performance than TANL, which was trained in a seq2seq fashion as well, using T5, with BART achieving

	CONLL04	NYT	DocRED	ADE
Strict Evaluation				
SpERT (Eberts and Ulges, 2020)	71.5†	-	-	79.2
Table-sequence (Wang and Lu, 2020)	73.6	-	-	80.1‡
JEREX (Eberts and Ulges, 2021)	-	-	40.4	-
TANL (Paolini et al., 2021)	71.4†	90.8	-	80.6
TANL (multi-dataset) (Paolini et al., 2021)	72.6†	90.5	-	80.0
REBEL	71.2	91.8	41.8	81.7
REBEL$_{pre-training}$	**75.4**	**92.0**	**47.1**	**82.2**
Boundaries Evaluation				
TPLinker (Wang et al., 2020)	-	91.9	-	-
REBEL$_{pre-training}$	-	**93.4**	-	-

Table 2: Comparison (Micro-F1) with most recent systems. † = explicit use of train+dev ‡ = filtered overlapping entities (2.8%)

	Precision	Recall	F1
CONLL04	75.59 ±1.53	75.12 ±0.64	75.35 ±1.01
NYT	91.71 ±0.10	92.21 ±0.14	91.96 ±0.07
DocRED	45.89 ±0.44	48.37 ±0.44	47.10 ±0.19
ADE	81.45 ±1.51	83.07 ±1.25	82.21 ±1.08
Re-TACRED	89.48 ±0.32	91.25 ±0.22	90.36 ±0.23

Table 3: Average micro metrics over 5 seeds (10-folds for ADE) for REBEL$_{pre-training}$. Standard deviation is indicated after the ± symbol.

	Precision	Recall	F1
CONLL04	75.22 ±1.30	69.01 ±1.68	71.97 ±1.00
NYT	91.50 ±0.12	92.02 ±0.11	91.76 ±0.04
DocRED	38.75 ±0.54	45.48 ±0.36	41.84 ±0.40
ADE	80.80 ±2.13	82.62 ±1.45	81.69 ±1.70
Re-TACRED	89.41 ±0.50	91.39 ±0.12	90.39 ±0.26

Table 4: Average micro metrics over 5 seeds for REBEL on test sets. Standard deviation is indicated after the ± symbol.

lower results for their approach. Therefore, our triplet linearization approach shows an improvement over other decoding strategies.

Results on RE for DocRED show that, despite being pre-trained on a sentence-based RE, REBEL can perform competitively on document-level RE, without the need for complex pipelines.

Moreover, by having a pre-trained version available, REBEL enables quick fine-tuning on newer domains, such as ADE, with different or fewer relation types, or including entity types. While in order to achieve the best performance we train for longer epochs, REBEL still needs fewer training steps to achieve competitive results compared to the other systems. For instance, Paolini et al. (2021) train CONLL04 for up to 200 epochs, Wang and Lu (2020) for up to 5,000, while our model needs less than 30 to achieve state-of-the-art results. Each of these systems uses large language models that can be expensive to train, and shorter training time can significantly decrease the costs.

5.2 Budget Training

We explore the training efficiency of REBEL$_{pre_trained}$, and show the performance when fine-tuned on a low number of epochs. We experiment with CONLL04 and NYT compared to the non-pre-trained model, SpERT and TANL. SpERT was trained for just 20 epochs on CONLL04, while TANL in its non-multi-dataset version is trained for 200 epochs. We adjust each learning rate scheduler to the number of epochs and re-train each model for different epochs and seeds.

Figures 2 and 3 show how in just 8 epochs for CONLL04 and 3 for NYT, REBEL$_{pre_trained}$ can achieve a similar performance as the previous state of the art. While the experiments are on the dev set,

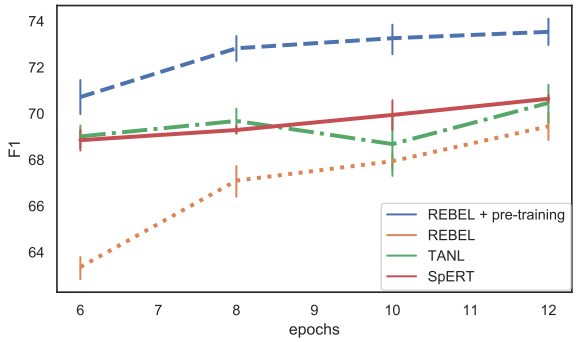

Figure 2: Micro-F1 performances on CONLL04 dev set averaged over 5 seeds.

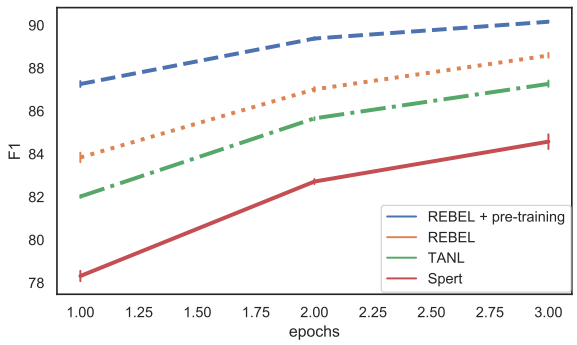

Figure 3: Micro-F1 performances on NYT dev set averaged over 3 seeds.

we do not observe big differences in performance between test and dev for these two datasets (see Appendix A.1 Tables 6 and 8). These results also highlight the importance of pre-training REBEL, as it achieves close to the final performance within a few epochs. Also note that while other models achieve lower performances, they also reach close to their final ones. Training for longer times and using early stopping on the validation performance are approaches used by most state-of-the-art models, but this can lead to long and expensive training times. Our experiments show that training for fewer epochs may lead to a small decrease in performance, but it brings the benefit of a more affordable training time. The comparison with other models should also take into account that our pre-trained approach has been previously trained on a massive dataset for 6 epochs, which combined with the fine-tuning in this experiment would lead to longer training times. However, all the other models also rely on pre-trained LM and, similarly, REBEL just needs to be pre-trained once and then quickly fine-tuned on these new datasets.

	F1
LUKE (Yamada et al., 2020)	90.3
RoBERTa$_{LARGE}$ + entity marker (Zhou and Chen, 2021)	**90.5**
REBEL	90.4
REBEL$_{pre-training}$	90.4

Table 5: Results on Re-TACRED

5.3 Relation Classification

As Table 5 shows, REBEL performs fairly well on RC despite being designed for RE. While Zhou and Chen (2021) presented a model with better results (91.1 F1) using entity types, we compare our models with those that do not use them. Both versions of REBEL achieve the same performance, in this case, in contrast to what we saw with RE. This may be due to the pre-training task being solely RE, as well as the size of the dataset.

For REBEL, we evaluate using free generation in the RC setup. Paolini et al. (2021) use likelihood-based prediction which leads to an increase in performance by computing the likelihood of each relation type to be decoded with the two given entities. However, this also leads to an overhead of computation for datasets with a high number of relations such as Re-TACRED. For this reason, we use free generation and are unable to compute results for Re-TACRED using TANL.

6 Conclusion

We have presented REBEL, alongside a new distantly supervised dataset for pre-training. REBEL frames RE into a seq2seq task and, by leveraging BART, achieves state-of-the-art performances in an array of RE benchmarks. We have also shown its flexibility in adapting to new domains, by training on just a few epochs to attain results that are comparable to the previous state of the art, as well as the possibility of using it to perform Relation Classification.

We make REBEL$_{pre-training}$ available as a standalone RE for more than 200 relation types together with a pre-trained RE model to serve as a baseline when fine-tuning on new RE datasets. Nonetheless, REBEL is based on BART-large, which has a big parameter footprint. Therefore, we also plan to release a pre-trained REBEL-base using BART-base. This will enable quick and efficient RE.

Moreover, our dataset creation pipeline enables a quick and effortless way of obtaining large high-

quality RE datasets in multiple languages from a Wikipedia dump. Since both Wikipedia and Wikidata are in constant change, our method provides a way to keep up with those changes and to have up-to-date RE datasets.

We leave to future work the possibility of using a multi-dataset approach as in Paolini et al. (2021), including both RE and RC datasets, and seeing if it retains or improves performance. Furthermore, using our silver dataset as pre-training could lead to improved performance for other systems, especially those which have shown better performance than REBEL without pre-training, such as Wang and Lu (2020) for CONLL04.

Acknowledgments

We would like to thank the authors of Elsahar et al. (2018) for the T-REx open code from which cRocoDiLe was built.

This research was funded by the European Union's H2020 Marie Skłodowska-Curie project *Knowledge Graphs at Scale* (KnowGraphs) under H2020-EU.1.3.1. (grant agreement ID: 860801).

References

Heike Adel and Hinrich Schütze. 2017. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1723–1729, Copenhagen, Denmark. Association for Computational Linguistics.

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online. Association for Computational Linguistics.

Giuseppe Attardi. 2015. Wikiextractor. https://github.com/attardi/wikiextractor.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12564–12573.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Rexhina Blloshmi, Simone Conia, Rocco Tripodi, and Roberto Navigli. 2021. Generating senses and roles: An end-to-end model for dependency- and span-based semantic role labeling. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3786–3793. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Rich Caruana. 1998. *Multitask Learning*, pages 95–133. Springer US, Boston, MA.

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems*, I-SEMANTICS '13, page 121–124, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with transformer pre-training. In *ECAI*, pages 2006–2013.

Markus Eberts and Adrian Ulges. 2021. An end-to-end model for entity-level relation extraction using multi-instance learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3650–3660, Online. Association for Computational Linguistics.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics*, 45(5):885–892. Text Mining and Natural Language Processing in Pharmacogenomics.

Arzoo Katiyar and Claire Cardie. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 917–928, Vancouver, Canada. Association for Computational Linguistics.

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. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. 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. 2019. Roberta: A robustly optimized bert pretraining approach.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8528–8535.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *9th International Conference on Learning Representations, ICLR 2021*.

Sachin Pawar, Pushpak Bhattacharyya, and Girish Palshikar. 2017. End-to-end relation extraction using neural networks and Markov Logic Networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 818–827, Valencia, Spain. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163, Berlin, Heidelberg. Springer Berlin Heidelberg.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

George Stoica, Emmanouil Antonios Platanios, and Barnabas Poczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13843–13850.

Bruno Taillé, Vincent Guigue, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. Let's Stop Incorrect Comparisons in End-to-end Relation Extraction! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3689–3701, Online. Association for Computational Linguistics.

Jue Wang and Wei Lu. 2020. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1706–1721, Online. Association for Computational Linguistics.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. TPLinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

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.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9507–9514. AAAI Press.

Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. 2019. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 367–377, Hong Kong, China. Association for Computational Linguistics.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. Minimize exposure bias of Seq2Seq models in joint entity and relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 236–246, Online. 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*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Suncong Zheng, Yuexing Hao, Dongyuan Lu, Hongyun Bao, Jiaming Xu, Hongwei Hao, and Bo Xu. 2017. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, 257:59–66. Machine Learning and Signal Processing for Big Multimedia Analysis.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212, Berlin, Germany. Association for Computational Linguistics.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *CoRR*, abs/2102.01373.

A Appendix

A.1 Results

Performances on the different dev sets can be found in Tables 6 and 8.

	Precision	Recall	F1
CONLL04	77.53 ±1.96	74.2 ±1.26	76.13 ±1.02
NYT	91.64 ±0.26	92.31 ±0.12	91.97 ±0.13
DocRED	46.65 ±0.94	49.19 ±0.43	47.89 ±0.68
Re-TACRED	89.59 ±0.21	90.81 ±0.25	90.19 ±0.13

Table 6: Average micro metrics over 5 seeds for REBEL$_{pre-training}$ on dev sets. Standard deviation is indicated after the ± symbol.

A.2 Reproducibility

Experiments were performed using a single NVIDIA 3090 GPU with 64GB of RAM and Intel® Core™ i9-10900KF CPU.

The hyperparameters were manually tuned on the validation sets for each dataset, but mostly left at default values for BART. The ones used for the final results can be found in Table 7. The number

	Max epochs	Learning Rate	Warm-up	Weight Decay	Batch size	Time per epoch
CONLL04	33	10^{-5}	10%	0.01	32	30 sec
NYT	42	$2.5 \cdot 10^{-5}$	10%	0.1	24	8 min
DocRED	20	10^{-5}	10%	0.01	32	2 min
ADE	25	10^{-5}	10%	0.01	32	1 min
Re-TACRED	6	10^{-5}	10%	0.01	32	8.5 min
REBEL	3	10^{-5}	1000 steps	0	32	9 hours

Table 7: Hyperparameters for the different datasets.

	Precision	Recall	F1
CONLL04	74.69 $\pm$0.76	71.66 $\pm$1.01	73.14 $\pm$0.73
NYT	91.44 $\pm$0.12	92.02 $\pm$0.15	91.72 $\pm$0.10
DocRED	46.27 $\pm$1.17	35.92 $\pm$1.81	40.40 $\pm$0.86
Re-TACRED	89.31 $\pm$0.20	90.87 $\pm$0.41	90.08 $\pm$0.19

Table 8: Average micro metrics over 5 seeds for REBEL on dev sets. Standard deviation is indicated after the $\pm$ symbol.

of parameters for REBEL is the same as for BART-large, 406M parameters, with a negligible increase from the newly added tokens.