90% F1 Score in Relational Triple Extraction: Is it Real ?

Pratik Saini and **Samiran Pal** and **Tapas Nayak** and **Indrajit Bhattacharya**
TCS Research, India
{pratik.saini,samiran.pal,nayak.tapas,b.indrajit}@tcs.com

Abstract

Extracting relational triples from text is a crucial task for constructing knowledge bases. Recent advancements in joint entity and relation extraction models have demonstrated remarkable F1 scores ($\geq$ 90%) in accurately extracting relational triples from free text. However, these models have been evaluated under restrictive experimental settings and unrealistic datasets. They overlook sentences with zero triples (zero-cardinality), thereby simplifying the task. In this paper, we present a benchmark study of state-of-the-art joint entity and relation extraction models under a more realistic setting. We include sentences that lack any triples in our experiments, providing a comprehensive evaluation. Our findings reveal a significant decline (approximately 10-15% in one dataset and 6-14% in another dataset) in the models' F1 scores within this realistic experimental setup. Furthermore, we propose a two-step modeling approach that utilizes a simple BERT-based classifier. This approach leads to overall performance improvement in these models within the realistic experimental setting.

1 Introduction

A crucial aspect of the relation extraction task involves the identification of sentences that lack any relational triples. This aspect naturally arises in real-world relation extraction scenarios. For instance, when extracting knowledge graph triples from online text, the majority of sentences may not mention any such triples. Although this aspect has been explored in other NLP tasks, such as machine reading comprehension, where models should correctly identify when a given passage lacks an answer rather than providing an incorrect one (Rajpurkar et al., 2018; Kundu and Ng, 2018; Sulem et al., 2021), it has not received sufficient attention in recent relation extraction research.

There are two distinct approaches for entity and relation extraction: Classification approach and joint approach. In the classification approach (Hoffmann et al., 2011; Zeng et al., 2014, 2015; Nayak and Ng, 2019; Jat et al., 2017), entities are already given and models focus on classifying the relations among pairs of entities. This approach includes sentences with zero triples in the experiments, where the relation among all entity pairs in such sentences is labeled as a 'None' relation. On the other hand, the joint extraction approach (Zeng et al., 2018; Takanobu et al., 2019; Nayak and Ng, 2020; Wei et al., 2020; Wang et al., 2020; Zheng et al., 2021; Li et al., 2021; Wei et al., 2020; Yan et al., 2021; Shang et al., 2022) involves models extracting both entities and relations simultaneously. However, in this approach, sentences with zero triples are not considered in the experiments, which makes the task significantly easier. Consequently, recent joint extraction models achieve exceptionally high F1 scores on benchmark datasets.

In this study, our objective is to assess the performance of state-of-the-art relational triples extraction models when sentences with zero triples are included. To achieve this, we conduct comprehensive experiments using the widely used New York Times (NYT) datasets. We evaluate a total of 9 recent state-of-the-art models in an end-to-end fashion. The results of our experiments reveal a significant decline in the performance of these models under this experimental setting. Across all of the evaluated models, we observe an approximate drop of 10-15% in the F1 score in one dataset, and a drop of around 6-14% in another dataset. These findings highlight the challenges posed by sentences without triples and emphasize the need for improved approaches to handle such cases effectively.

Additionally, we have identified that sentences often contain clue tokens that can be leveraged to detect the presence of relations, even without identifying the corresponding entities. We include such examples in Table 1 for illustrations. Building upon this observation, we introduce a BERT-based

Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP, pages 1–11
December 6, 2023 ©2023 Association for Computational Linguistics

Sentence	Triples
Paul Allen , a co-founder of Microsoft , paid the bills for aircraft designer Burt Rutan to develop SpaceShipOne , the craft that won the $ 10 million Ansari X Prize last year for reaching suborbital space .	Microsoft ; Paul Allen ; /business/company/founders Paul Allen ; Microsoft ; /business/person/company
But Schaap seems as comfortable in that role as Joe Buck , the Fox baseball and football sportscaster who so clearly benefited from learning beside his father , Jack Buck , the late voice of the St. Louis Cardinals .	Jack Buck ; Joe Buck ; /people/person/children

Table 1: Examples of relation clue tokens (in Pink) for determining the presence of a relation in the sentences.

zero-cardinality classifier (ZCC) model that effectively filters out sentences with zero triples. We explore both binary classification and multi-class multi-label (MCML) classification approaches for this purpose. To tackle the task at hand, we propose a two-step modeling approach. In the first step, we employ the ZCC model to classify the sentences, determining whether they contain zero triples or not. In the second step, we utilize the outputs of the ZCC model to guide the 9 state-of-the-art triples extraction models, effectively solving the task. Notably, our experimental results demonstrate that this two-step approach outperforms or achieves competitive performance compared to end-to-end modeling in this novel setting of the task. Furthermore, it offers advantages in terms of training time for the models[1].

2 End-to-End Modeling of Relation Extraction with Zero-Cardinality

For our experiments, we select nine state-of-the-art joint entity and relation extraction models: PtrNet (Nayak and Ng, 2020), TPLinker (Wang et al., 2020), CasRel (Wei et al., 2020), TDEER (Li et al., 2021), PRGC (Zheng et al., 2021), PFN (Yan et al., 2021), GRTE (Ren et al., 2021), OneRel (Shang et al., 2022), and BiRTE (Ren et al., 2022). All of these models utilize BERT (Devlin et al., 2019) as an encoder. For our experiments with the NYT24* dataset, where sentences are cased, we utilize the BERT_base_cased model. On the other hand, for the NYT29* dataset, where sentences are uncased, we use the BERT_base_uncased model.

PtrNet (Nayak and Ng, 2020) adopts a sequence-to-sequence (seq2seq) approach, extracting triples uniformly regardless of the relations involved. The remaining models employ relation-specific sequence or matrix labeling methods to extract triples. Originally, these models are trained solely on sentences containing one or more triples, excluding sentences with zero triples from their training and test datasets. However, we adapt these models to handle sentences with zero triples as well. In the case of sequence labeling or matrix labeling approaches, all tokens in the zero-cardinality sentences are labeled with the ’O’ tag (representing the "other" tag). For sequence generation approaches (such as seq2seq), the decoder generates the "end of sequence" (EOS) tag as the first token, indicating the absence of any relational triple in the sentence.

Below is a brief description of each of these models. We employ the same hyper-parameters as specified in their respective papers.

2.1 PtrNet (Nayak and Ng, 2020)

This model utilizes a seq2seq framework with pointer network-based decoding for joint entity and relation extraction. Each triple is represented by the start and end indices of the subject and object entities in the sentence, along with the corresponding relation class label. To generate the complete triple, their decoding framework extracts four indexes at each time step, capturing the subject and object entities as well as the relation between them. This enables the model to incrementally construct the entire triple. For a fair comparison with other state-of-the-art (SOTA) models, the original BiLSTM encoder is replaced with BERT, a powerful language representation model. This integration of BERT into the model ensures compatibility and consistency with the advancements in the field, allowing for more accurate and robust results.

2.2 CasRel (Wei et al., 2020)

CasRel employs a two-stage extraction process for relation extraction. In the first stage, it utilizes a 0/1 tagging scheme to identify all subject entities present in the text. This initial stage focuses on accurately identifying and labeling the subject entities involved in the relations. In the subsequent

[1] Any code or data related to this paper will be made available at https://github.com/pratiksaini4/ZeroCardinalityImpactOnRE.

stage, for each subject entity and for each relation, CasRel applies another round of 0/1 tagging to identify the corresponding object entities. This object tagging process is iterative and carried out sequentially for each subject entity. By performing this iterative tagging approach, CasRel ensures comprehensive identification of the object entities associated with each subject entity, enabling a more precise extraction of relational triples.

2.3 TPLinker (Wang et al., 2020)

TPLinker also adopts a sequence labeling approach for the relation extraction task. However, to effectively address the challenges posed by overlapping triples, it employs a separate sequence labeling process for each relation. To link the tokens within the sentence, TPLinker utilizes a handshaking tagging scheme. It constructs a matrix representing the tokens in the sentence, where the rows and columns correspond to the tokens. The handshaking tags are employed to establish connections between tokens. Initially, TPLinker identifies all entities in the sentence using the 'EH-ET' (entity head to entity tail) tag. In the matrix, a cell with a value of 1 indicates that the token in the corresponding row represents the start of an entity, while the token in the column represents the end of the entity. Additionally, TPLinker employs two other handshaking tags, namely SH-OH' (subject head to object head) and ST-OT' (subject tail to object tail). These tags are used to link the subject and object entities for each specific relation. Separate matrices are tagged for each relation using these handshaking tags. By applying this approach, TPLinker effectively links the subject and object entities for each relation, enabling accurate extraction of relational triples. The initial set of entities obtained from the 'EH-ET' tagging stage serves to filter out unwanted triples extracted during the relation-specific tagging stage.

2.4 TDEER (Li et al., 2021)

This task employs a multi-stage sequence labeling approach. In the initial stage, a 0/1 tagging scheme is utilized to extract subject and object entities. Additionally, a multi-label classification technique is employed to identify all possible relations present in the sentence. In the subsequent stage, for each subject entity and relation pair, the start position of the corresponding object entity is identified. If this start position aligns with any of the object entities extracted in the first stage, the triple is considered valid and retained. Conversely, if no match is found, the triple is deemed invalid and discarded. This rigorous validation process ensures the accuracy and reliability of the extracted triples.

2.5 PRGC (Zheng et al., 2021)

In this model, the first step involves identifying a set of potential relations within the sentence, as well as establishing a global correspondence matrix between the subject and object entities. In the subsequent stage, relation-specific sequence taggers are employed to label the subject and object entities accordingly. These taggers provide fine-grained annotations, enabling precise identification of the entities involved. Finally, the global correspondence matrix is utilized to make informed decisions regarding which triples to accept or discard. By considering the interplay between the subject and object entities and their respective relations, the model ensures the selection of valid and meaningful triples while discarding any irrelevant or incorrect ones.

2.6 GRTE (Ren et al., 2021)

This approach utilizes a table filling method where separate tables are maintained for each relation. Each cell in the table represents whether a token pair is associated with the corresponding relation or not. These tables are populated using local features or the historical information of a limited number of token pairs. GRTE enhances the table-filling by incorporating two types of global features. The first type pertains to the global association of entity pairs, while the second type focuses on relations. GRTE initially generates a table feature for each relation. Subsequently, these table features for all relations are combined, resulting in the creation of a subject-related global feature and an object-related global feature. These global features are then utilized iteratively to refine the individual table features. By employing this refined table-filling approach, all triples can be extracted based on the information stored in the populated tables. This method enables the accurate and comprehensive extraction of relational triples.

2.7 PFN (Yan et al., 2021)

The model consists of two main modules: the Named Entity Recognition (NER) module and the Relation Extraction (RE) module. In the NER module, all named entities in the sentences are extracted, capturing their complete spans. This module focuses on identifying and delineating entities present in the text. The RE module operates separately for each relation. It employs matrix labeling techniques to identify the starting tokens of subject and object entity pairs. The full span of these entities is obtained from the entities previously identified by the NER module. By leveraging the information provided by the NER module, the RE module can accurately determine the boundaries and positions of the subject and object entities for each relation.

2.8 OneRel (Shang et al., 2022)

The approach utilized in this task is a relation-specific horns-tagging method. For each relation in the set of relations, a matrix is maintained, consisting of four types of tags: 'HB-TB', 'HB-TE', 'HE-TE', and 'O'. Here, 'H/T' represents the head or tail entity, while 'B/E' denotes the beginning and ending of an entity. The rows of this matrix correspond to the head entity tokens, while the columns correspond to the tail entity tokens derived from the source text. Following the tagging of these matrices, a scoring-based classifier is employed to iterate through all possible combinations and discard triples with low confidence scores. This process enables the identification and retention of high-quality triples based on their associated confidence scores.

2.9 BiRTE (Ren et al., 2022)

This model employs a multi-stage bidirectional tagging-based mechanism. In the initial stage, the model focuses on identifying subject and object entities. Subsequently, in the second stage, it further refines the identification of object entities based on the previously identified subject entities, and vice versa. Finally, in the last stage, subject-object pairs are classified based on their respective relations. All these stages are trained together as a single model, ensuring a comprehensive and integrated approach to relation extraction.

3 Two-step Modeling of Relation Extraction with Zero-Cardinality

We have observed that most relational triples in sentences are associated with specific clue tokens. While this may not always hold true due to the distant supervision used in creating the NYT datasets, it is applicable to many cases. We have included relevant examples in Table 1. Based on this observation, we aim to investigate whether a BERT-based classification model can learn to identify the presence of relational triples in these sentences using the clue tokens, without requiring knowledge of the specific entities involved in the triples.

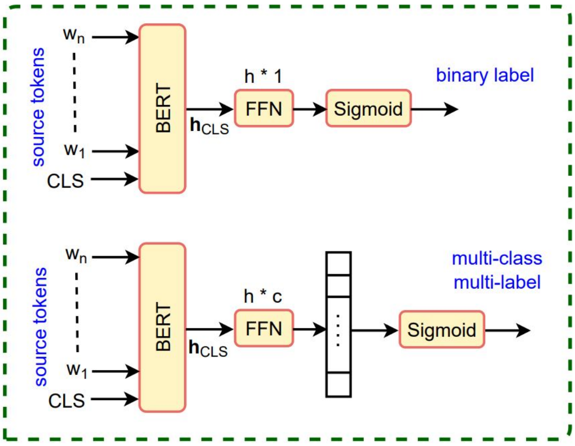

Figure 1: Architecture of our zero-cardinality classifier. c is the number of relations.

To accomplish this, we feed the sentences with a 'CLS' prefix token (CLS w_1 w_2 ..... w_n) into a pre-trained BERT_base model with a hidden dimension of h. We utilize the vector representation of the 'CLS' token to determine whether the sentence contains any relational triples or not. We refer to this classifier as the zero-cardinality classifier (ZCC).

We explore two distinct approaches for this classifier:

(i) The first approach involves binary classification to determine whether a sentence contains any triples or not. However, in this approach, we do not explicitly utilize the set of relations.

(ii) The second approach employs a multi-class multi-label (MCML) classification, which focuses on identifying the specific relations within the relation set. Sentences without any triples are assigned no positive labels.

To begin, we train the classifier on the 'WZ' training dataset, while training the joint extraction models on the 'NZ' training set. During the inference phase, if the classifier model indicates the presence of triples in a test instance, we subsequently pass it to the joint extraction models to extract the exact triples. This two-step process enables us to effectively filter out sentences that do not contain any triples.

We include the architecture of our proposed zero-cardinality classifier in Fig 1. We use binary cross-entropy loss and AdamW (Loshchilov and Hutter, 2019) optimizer to update the model parameters. We use mini-batch size of 16 and an early stop

	NYT24*			NYT29*		
	Train	Validation	Test	Train	Validation	Test
#sentences with >=1 triples	56,196	5,000	5,000	63,306	7,033	4,006
#triples in above sentences	88,366	8,489	8,120	78,973	8,766	5,859
#sentences with zero triples	145,767	4,969	4,969	177,861	4,940	4,601

Table 2: The statistics of the NYT24* and NYT29* datasets.

criterion during training. Our experiments have demonstrated that this two-step approach significantly enhances the overall performance of the joint models on the test set, encompassing sentences both with and without triples.

4 Datasets Preparation & Evaluation Metric

The New York Times (NYT) dataset holds significant importance as a benchmark for relation extraction. Several studies (Zeng et al., 2018; Takanobu et al., 2019; Nayak and Ng, 2020) utilize the derived NYT29 and NYT24 datasets, which originate from the original NYT10 (Riedel et al., 2010) and NYT11 (Hoffmann et al., 2011) training corpus, respectively. Zeng et al. (2018); Takanobu et al. (2019); Nayak and Ng (2020) exclude sentences without triples and partition the dataset into training, validation, and test sets. Subsequent research papers (Wei et al., 2020; Wang et al., 2020; Zheng et al., 2021; Li et al., 2021; Wei et al., 2020; Yan et al., 2021; Shang et al., 2022) build upon this modified version of the datasets, which is comparatively easier, and achieve exceptionally high F1 scores on these datasets. This trend reflects the prevalence of simplified datasets in recent works, potentially overestimating the performance of relation extraction models when faced with more realistic scenarios.

In order to enhance the realism of the joint extraction task, we augment the NYT29 and NYT24 datasets by incorporating sentences with zero triples from the original NYT10 and NYT11 training corpus, respectively. These augmented datasets are referred to as NYT29* and NYT24* hereafter. The specifics regarding the training, validation, and test splits of the NYT24* and NYT29* datasets can be found in Table 2.

To evaluate the state-of-the-art (SOTA) models, we conduct experiments using two distinct training and test settings. These settings are as follows:

(i) **NoZero (NZ)**: In this setting, only sentences containing one or more triples are included for training and testing purposes.

(ii) **WithZero (WZ)**: This setting encompasses the sentences from the NZ set, along with additional sentences with zero triples from the corresponding original NYT datasets.

By employing these two different experimental designs, we aim to gain insights into the robustness of the joint extraction models and their ability to handle different scenarios.

4.1 Evaluation Metric

For evaluating the performance of the state-of-the-art (SOTA) models, we employ triple-level precision, recall, and F1 score as the evaluation metrics. In order to determine the correctness of an extracted triple, we compare it with the ground truth triple. A triple is considered correct if both the corresponding entities and the relation match accurately. In the case of an 'Exact' match, we require the full span of the entities to match precisely, as specified in the respective papers. However, in the case of a 'Partial' match, we only compare the first or last token of the entities with the ground truth.

5 Results & Discussion

To begin our analysis, we assess the performance of state-of-the-art (SOTA) end-to-end models under the new experiment settings, which now include sentences with zero cardinality. The results of these experiments are presented in Table 3. Initially, we train these models solely on the 'NZ' sentences and evaluate their performance on both the 'NZ' and 'WZ' sentences. Upon evaluation, we observe a significant decline in the F1 score on the WZ' sentences compared to the NZ' sentences. Across the NYT24* and NYT29* datasets, the F1 score experiences a decrease of approximately 14-24%. Furthermore, the precision score for all these models exhibits a sharp drop, as they extract triples from sentences that do not contain any triples. This outcome is expected since the models have not been exposed to any examples featuring zero triples during the training phase.

Next, we proceed to train these models using the

		Test setting →	NZ			WZ				
	Training setting ↓	Model	Prec.	Rec.	F1	Prec.	Rec.	F1	% point ↓	
NYT24*	NZ	OneRel	0.926	0.918	0.922	0.678	0.918	0.780	14.2	
		BiRTE	0.914	0.920	0.917	0.628	0.920	0.747	17.0	
		TDEER	0.922	0.908	0.915	0.644	0.908	0.754	16.1	
		PRGC	0.918	0.884	0.901	0.670	0.884	0.762	13.9	
		GRTE	0.929	0.924	0.926	0.645	0.924	0.760	16.6	
		PtrNet	0.898	0.894	0.896	0.538	0.894	0.671	22.5	
		CasRel	0.894	0.890	0.892	0.612	0.890	0.725	16.7	
		TPLinker	0.913	0.917	0.915	0.643	0.917	0.756	15.9	
		PFN*	0.892	0.919	0.905	0.557	0.919	0.694	21.1	
	WZ	OneRel	0.926	0.773	0.843	0.828	0.773	0.800	4.3	**12.2**
		BiRTE	0.898	0.858	0.878	0.786	0.858	0.820	5.8	**9.7**
		TDEER	0.914	0.905	0.909	0.637	0.905	0.748	16.1	**16.7**
		PRGC	0.905	0.777	0.836	0.791	0.777	0.784	5.2	**11.7**
		GRTE	0.920	0.769	0.838	0.824	0.769	0.796	4.2	**13.0**
		PtrNet	0.932	0.697	0.798	0.838	0.697	0.761	3.7	**13.5**
		CasRel	0.915	0.878	0.896	0.643	0.878	0.742	15.4	**15.0**
		TPLinker	0.923	0.808	0.861	0.823	0.807	0.815	4.6	**10.0**
		PFN*	0.910	0.732	0.812	0.804	0.732	0.766	4.6	**13.9**
NYT29*	NZ	OneRel	0.805	0.726	0.763	0.528	0.726	0.611	15.2	
		BiRTE	0.794	0.724	0.757	0.484	0.724	0.580	17.7	
		TDEER	0.813	0.707	0.756	0.530	0.707	0.606	15.0	
		PRGC	0.807	0.701	0.750	0.509	0.701	0.590	16.0	
		GRTE	0.804	0.726	0.763	0.492	0.726	0.587	17.6	
		PtrNet	0.790	0.710	0.748	0.394	0.710	0.507	24.1	
		CasRel	0.795	0.712	0.751	0.488	0.712	0.579	17.2	
		TPLinker	0.805	0.718	0.759	0.456	0.718	0.558	20.1	
		PFN*	0.777	0.720	0.748	0.474	0.720	0.572	17.6	
	WZ	OneRel	0.841	0.657	0.738	0.755	0.657	0.703	3.5	**6.0**
		BiRTE	0.833	0.663	0.738	0.698	0.663	0.680	5.8	**7.7**
		TDEER	0.788	0.708	0.746	0.536	0.708	0.611	13.5	**14.5**
		PRGC	0.842	0.639	0.727	0.755	0.639	0.692	3.5	**5.8**
		GRTE	0.840	0.624	0.716	0.759	0.623	0.684	3.2	**7.9**
		PtrNet	0.876	0.620	0.726	0.720	0.620	0.666	6.0	**8.2**
		CasRel	0.807	0.708	0.754	0.541	0.708	0.613	14.1	**13.8**
		TPLinker	0.775	0.636	0.698	0.686	0.636	0.660	3.8	**9.9**
		PFN*	0.833	0.600	0.697	0.748	0.600	0.666	3.1	**8.2**

Table 3: Performance of the joint extraction models in the end-to-end approach on the NYT24* and NYT29* datasets with different train/test settings. * marked models are evaluated using partial entity matching as per their paper. F1 score in green color are the results obtained without zero-cardinality sentences. F1 score in red color are the results obtained with zero-cardinality sentences. The % point ↓ numbers in **bold** are the difference between the F1 scores in green and red.

'WZ' sentences. Upon analysis, we note that their performance on the 'NZ' sentences experiences a decline of 4-8%, with the exception of the TDEER and CasRel models. Interestingly, the TDEER and CasRel models exhibit comparable performance on the 'NZ' test set, regardless of whether they were trained on 'NZ' or 'WZ' training data. However, the introduction of sentences with zero triples during the training process tends to confuse these models, leading to a negative impact on their recall. Consequently, the models struggle to accurately extract valid triples due to the presence of such adversarial examples. Furthermore, in this training setting, we observe an improvement of 2-8% in the models' performance on 'WZ' sentences. Nevertheless, the best F1 score reported on the stringent 'NZ' test set for NYT24* is 0.926 (achieved by the GRTE model). In contrast, the best F1 score attained on the 'WZ' test set for NYT24* is 0.82 (achieved by the BiRTE model). This signifies a 10% drop in the best F1 score when transitioning to the experiment's more diverse setting. Similarly, we observe a 6% decrease in the best achieved F1 scores on the 'WZ' test set for NYT29* compared to the 'NZ' test set.

Next, we delve into the analysis of the impact of our proposed two-step approach for this task. The first step involves utilizing the zero-cardinality classifier to predict sentences with zero cardinality, i.e., sentences that either contain triples or do not. The performance of the classification model using both binary and multi-class multi-label (MCML)

	NYT24*			NYT29*		
	Prec.	Rec.	F1	Prec.	Rec.	F1
ZCC_{binary}	0.887	0.867	0.877	0.801	0.888	0.842
ZCC_{MCML}	0.881	0.884	0.883	0.823	0.824	0.823

Table 4: Performance of the zero cardinality classifier (ZCC) model on NYT24* and NYT29* datasets in the binary classification and multi-class multi-label classification (MCML) settings.

	NYT24*				NYT29*			
	multi-class multi-label				binary			
Model	Prec.	Rec.	F1	% ↑	Prec.	Rec.	F1	% ↑
OneRel	0.832	0.836	0.834	3.43	0.740	0.664	0.700	-0.27
BiRTE	0.819	0.839	0.829	0.85	0.679	0.663	0.671	-0.95
TDEER	0.830	0.830	0.830	8.23	0.749	0.649	0.696	8.52
PRGC	0.822	0.811	0.816	3.26	0.744	0.645	0.691	-0.14
GRTE	0.835	0.842	0.839	4.30	0.740	0.661	0.699	1.41
PtrNet	0.806	0.815	0.811	4.95	0.677	0.650	0.663	-0.33
CasRel	0.807	0.812	0.810	6.73	0.676	0.653	0.665	5.13
TPLinker	0.816	0.839	0.828	1.23	0.681	0.656	0.668	0.81
PFN*	0.805	0.833	0.818	5.20	0.726	0.658	0.690	2.42

Table 5: Performance of the SOTA models in the two-step modeling on the relational triple extraction task with zero-cardinalty sentences. At the first-step, we use multi-class multi-label classification for NYT24* dataset and binary classification for NYT29* dataset.

classification is provided in Table 4. The classification model was trained on 'WZ' sentences for both the NYT24* and NYT29* datasets. Both binary classification and multi-class multi-label classification demonstrate competitive performance on both datasets. Multi-class multi-label classification exhibits slightly higher performance on the NYT24* dataset, while binary classification yields marginally better results on the NYT29* dataset.

In the second step of our two-step approach, only the sentences predicted by the classification model to have existing triples are passed on to the triple extraction model. For this step, we train the state-of-the-art (SOTA) models exclusively on the 'NZ' sentences to facilitate triple extraction. In Table 5, we present the comprehensive performance evaluation of the state-of-the-art (SOTA) model using the two-step approach for the triple extraction task. For the NYT24* dataset, we utilize the multi-class multi-label classifier, while for the NYT29* dataset, we employ the binary classification approach for zero-cardinality prediction.

Our observations reveal an improvement of approximately $\sim 8\%$ in the 'WZ' sentences for both the NYT24* and NYT29* datasets when employing the two-step approach compared to the end-to-end approach. Specifically, for the NYT24* dataset, all SOTA models exhibit enhanced performance with the two-step approach over the end-to-end approach. However, for the NYT29* dataset, the performance is not consistently improved. In the case of four models (OneRel, BiRTE, PRGC, and PtrNet), we observed a minor drop of up to $\sim 1\%$ with the two-step approach.

Overall, we conclude that the two-step approach either improves the performance of these models or achieves competitive performance when compared to the end-to-end approach in this new experimental setting for relation extraction.

5.1 Training Time of the Models

Table 6 presents the training time of various models used in our experiments. All training was conducted on an NVIDIA A100 GPU with 20 GB GPU memory. Our two-step approach for the relation extraction task in this new setting offers advantages over the end-to-end approach.

The training time for the SOTA models solely using 'NZ' data is considerable, primarily due to their utilization of BERT as the sentence encoder. However, when we incorporate sentences with zero triples in the training process (which account for almost three times the number of sentences with triples, as shown in Table 2), the training time sig-

	NYT24*		NYT29*	
	NZ	WZ	NZ	WZ
Onrel	18.33	68.35	21.05	83.06
BiRTE	6.67	25.49	6.24	32.74
TDEER	43.51	50.85	45.11	63.68
PRGC	20.25	56.70	18.96	62.87
GRTE	20.70	65.08	21.87	80.51
PtrNet	17.17	41.03	12.24	24.30
CasRel	18.23	65.20	20.54	75.95
TPLinker	26.19	122.65	43.40	168.91
PFN*	22.40	317.18	188.68	393.35
ZCC$_{binary}$	-	14.67	-	25.55
ZCC$_{MCML}$	-	14.49	-	25.65

Table 6: Training time of the models. First 9 rows are avg. training epoch time (in minutes) of five SOTA models on the 'NZ' and 'WZ' training data. Last two rows are avg. training time of the zero cardinality classification (ZCC) models with WZ training data.

nificantly increases for all models (refer to Table 6).

On the contrary, the zero-cardinality classifier only needs to be trained once for all models, resulting in substantial time savings. Additionally, training the zero-cardinality classifier itself is relatively quick due to its simple architecture.

6 Related Work

Extracting relational triples from text is a crucial task for constructing new knowledge bases or enhancing existing ones. In their efforts to address this task, Mintz et al. (2009); Riedel et al. (2010); Hoffmann et al. (2011) employed feature-based classification models. More recently, Zeng et al. (2014, 2015) utilized CNN models, which automatically extract features, for this purpose. Shen and Huang (2016); Jat et al. (2017); Nayak and Ng (2019) incorporated attention mechanisms into their models to enhance performance. Approaches such as Surdeanu et al. (2012); Lin et al. (2016); Vashishth et al. (2018) adopted a multi-instance relation extraction setting, where multiple sentences are used to capture features associated with a pair of entities. These approaches assume that entities have already been identified and focus solely on classifying relations between entity pairs.

Katiyar and Cardie (2016); Miwa and Bansal (2016); Bekoulis et al. (2018); Nguyen and Verspoor (2019); Nayak and Ng (2020) tried to bring the named entity recognition task and relation classification task together. Zheng et al. (2017) used a sequence tagging scheme to jointly extract the entities and relations. Zeng et al. (2018); Nayak and Ng (2020) proposed an encoder-decoder model to extract relational triples with overlapping entities. Takanobu et al. (2019) proposed a joint extraction model based on hierarchical reinforcement learning (HRL).

With the introduction of pre-trained models such as BERT (Devlin et al., 2019), many models used such models as sentence encoder to improve their performance. Models such as TPLinker (Wang et al., 2020), CasRel (Wei et al., 2020), TDEER (Li et al., 2021), PRGC (Zheng et al., 2021), PFN (Yan et al., 2021), GRTE (Ren et al., 2021), OneRel (Shang et al., 2022), and BiRTE (Ren et al., 2022) use BERT_base (Devlin et al., 2019) as an encoder and proposed table-filling method or relation specific tagging mechanism for joint entity and relation extraction. These models show remarkable performance on the NYT datasets in the restrictive experimental setting without considering the zero-cardinal sentences.

7 Conclusion

In this work, we present an innovative and challenging experiment design for relation extraction, which incorporates sentences containing zero triples (referred to as zero-cardinal sentences) in the dataset. We conduct comprehensive experiments involving 9 state-of-the-art (SOTA) models using the widely-used New York Times datasets. To tackle this task, we devise both an end-to-end modeling approach and a two-step modeling approach.

During our investigations, we make a significant observation in the end-to-end modeling, where we notice a drop in the F1 score by approximately 10-15% and 6-14% in two versions of the NYT dataset. To address this issue, we propose the integration of a BERT-based classifier as an additional step for this task. Remarkably, this approach either achieves performance comparable to the end-to-end approach or even surpasses it.

We believe that our benchmark, focusing on relational triple extraction with zero-cardinality, will prove immensely valuable for future research in this domain. By introducing this unique experiment design, we aim to stimulate further advancements and foster progress in this field.

8 Limitations

One limitation of this work is that we benchmark this task using 9 SOTA joint models. There are many other SOTA models published in this area but it is difficult to benchmark all of them. We chose the 9 models in such as way that different kind of design choices in these models are represented in our study. We chose Seq2Seq model (Nayak and Ng, 2020), horn tagging-based models (Wang et al., 2020; Shang et al., 2022), 0/1 tagging-based models (Wei et al., 2020; Li et al., 2021), table-filling models (Ren et al., 2021) for rigorous study of this area.

9 Ethics Statements

Our work does not have any ethical concerns.

References

Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*.

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

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*.

Sharmistha Jat, Siddhesh Khandelwal, and Partha Talukdar. 2017. Improving distantly supervised relation extraction using word and entity based attention. In *AKBC*.

Arzoo Katiyar and Claire Cardie. 2016. Investigating LSTMs for joint extraction of opinion entities and relations. In *ACL*.

Souvik Kundu and Hwee Tou Ng. 2018. A nil-aware answer extraction framework for question answering. In *EMNLP*.

Xianming Li, Xiaotian Luo, Cheng Jie Dong, Daichuan Yang, Beidi Luan, and Zhen He. 2021. TDEER: An efficient translating decoding schema for joint extraction of entities and relations. In *EMNLP*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

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

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *ACL*.

Tapas Nayak and Hwee Tou Ng. 2019. Effective attention modeling for neural relation extraction. In *CoNLL*.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *AAAI*.

Dat Quoc Nguyen and Karin Verspoor. 2019. End-to-end neural relation extraction using deep biaffine attention. In *ECIR*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *ACL*.

Feiliang Ren, Longhui Zhang, Shujuan Yin, Xiaofeng Zhao, Shilei Liu, Bochao Li, and Yaduo Liu. 2021. A novel global feature-oriented relational triple extraction model based on table filling. In *EMNLP*.

Feiliang Ren, Longhui Zhang, Xiaofeng Zhao, Shujuan Yin, Shilei Liu, and Bochao Li. 2022. A simple but effective bidirectional framework for relational triple extraction. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML and KDD*.

Y. Shang, Heyan Huang, and Xian-Ling Mao. 2022. OneRel: Joint entity and relation extraction with one module in one step. In *AAAI*.

Yatian Shen and Xuanjing Huang. 2016. Attention-based convolutional neural network for semantic relation extraction. In *COLING*.

Elior Sulem, Jamaal Hay, and Dan Roth. 2021. Do we know what we don't know? studying unanswerable questions beyond SQuAD 2.0. In *EMNLP*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP and CoNLL*.

Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *AAAI*.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. Reside: Improving distantly-supervised neural relation extraction using side information. In *EMNLP*.

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 *COLING*.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A novel cascade binary tagging framework for relational triple extraction. In *ACL*.

Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. 2021. A partition filter network for joint entity and relation extraction. In *EMNLP*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *EMNLP*.

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

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 *ACL*.

Heng Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Ming Xu, and Yefeng Zheng. 2021. PRGC: Potential relation and global correspondence based joint relational triple extraction. In *ACL*.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *ACL*.

A Appendix

A.1 GenBench Evaluation Cards

Motivation			
Practical	*Cognitive*	*Intrinsic*	*Fairness*
○			

Generalisation type					
Compositional	*Structural*	*Cross Task*	*Cross Language*	*Cross Domain*	*Robustness*
				○	

Shift type			
Covariate	*Label*	*Full*	*Assumed*
	○		

Shift source			
Naturally occuring	*Partitioned natural*	*Generated shift*	*Fully generated*
○			

Shift locus			
Train–test	*Finetune train–test*	*Pretrain–train*	*Pretrain–test*
○			

Table 7: We characterise all our experiments of Section 5 (○) in this datacard.