Effective Modeling of Generative Framework for Document-level Relational Triple Extraction

Pratik Saini and **Tapas Nayak**
TCS Research, India
{pratik.saini, nayak.tapas}@tcs.com

Abstract

Document-level relational triple extraction (DocRTE) is a complex task that involves three key sub-tasks: entity mention extraction, entity clustering, and relational triple extraction. Past work has applied discriminative models to address these three sub-tasks, either by training them sequentially in a pipeline fashion or jointly training them. However, while end-to-end discriminative or generative models have proven effective for sentence-level relational triple extraction, they cannot be trivially extended to the document level, as they only handle relation extraction without addressing the remaining two sub-tasks, entity mention extraction or clustering. In this paper, we propose a three-stage generative framework leveraging a pre-trained BART model to address all three tasks required for document-level relational triple extraction. Tested on the widely used DocRED dataset, our approach outperforms previous generative methods and achieves competitive performance against discriminative models.

1 Introduction

Extracting relational triples—composed of a subject entity, an object entity, and the relation between them—from documents is a vital yet challenging task in natural language processing (NLP). Unlike sentence-level relation extraction tasks (Zheng et al., 2017; Zeng et al., 2018; Nayak and Ng, 2020), the challenges in document-level extraction increase significantly. The first major challenge is the extended context of documents, which requires capturing long-distance dependencies between entities across larger spans of text. Another challenge is that an entity may appear multiple times in a document with different surface forms, making entity resolution crucial. This complexity is less pronounced in sentence-level tasks, where entities are generally mentioned only once within a shorter context. An example of document-level relational triple extraction (DocRTE) is shown in Table 1 to demonstrate the complexity of this task.

Generative models (Zeng et al., 2018; Nayak and Ng, 2020) have shown strong performance in sentence-level relational triple extraction. Building on this, Cabot and Navigli (2021) proposed REBEL for document-level relational triple extraction using the DocRED dataset (Yao et al., 2019). REBEL introduced a linearization scheme that encodes all triples in a document as a sequence of tokens, using BART-large as the base model. The model's decoder then generates a token sequence, from which triples are extracted through straightforward post-processing. However, REBEL's linearization scheme does not fully address the entity mention extraction and entity clustering sub-tasks within DocRTE. It only captures the initial mentions of entities involved in relations and does not handle the extraction of mentions with different surface forms. In contrast, Giorgi et al. (2022) proposed an alternative linearization scheme for DocRTE, including all mentions of subject and object entities in the output sequence for each relational triple. This approach partially addresses some challenges of mention extraction and entity clustering. However, it redundantly extracts entity clusters multiple times if they are involved in multiple relational triples, which increases sequence length without added value. It also overlooks clusters that do not appear in any relation.

In contrast, JEREX (Eberts and Ulges, 2021) proposed a three-stage approach to address the tasks of entity mention extraction, entity clustering, and relational triple extraction, which can be trained either in a pipeline or jointly. The first stage employs a span-based classifier to identify entity mentions within the document. The second stage uses pairwise classification between entity mentions for clustering, and the third stage applies a relation classifier to determine relationships, or 'no relation,' between pairs of entity clusters. TAG (Zhang

Proceedings of the Generative AI and Knowledge Graph Workshop (GenAIK), pages 1–12
January 19, 2025.

Document: Washington Place (William Washington House) is one of the first homes built by freed slaves after the Emancipation Proclamation of 1863 in Hampshire County , West Virginia , United States . Washington Place was built by William and Annie Washington in north Romney between 1863 and 1874 on land given to Annie by her former owner , Susan Blue Parsons of Wappocomo plantation . William Washington later acquired other properties on the hills north of Romney along West Virginia Route 28 and became the first African - American land developer in the state of West Virginia . One of his subdivisions is the " Blacks Hill " neighborhood of Romney , adjacent to the Washington Place homestead . Washington Place was bought and restored by Ralph W. Haines , a local attorney and historic preservationist .	
Entity Clusters: C1: (['Washington Place', 'William Washington House','Washington Place', 'Washington Place', 'Washington Place'], 'LOC'), C2: (['Emancipation Proclamation'], 'MISC'), C4: (['Hampshire County'], 'LOC'), C5: (['West Virginia', 'West Virginia'], 'LOC'), C6: (['United States'], 'LOC'), C7: (['William', 'William Washington'], 'PER'),C8: (['Annie Washington', 'Annie'], 'PER'), C9: (['Romney', 'Romney', 'Romney'], 'LOC'), C12: (['Susan Blue Parsons'], 'PER'), C13: (['Wappocomo plantation'], 'LOC'), C15: (['West Virginia Route 28'], 'LOC'), C18: (['Blacks Hill'], 'MISC'), C19: (['Ralph W. Haines'], 'PER')	
Relational Triples: ['C2', 'C6', 'country'], ['C4', 'C5', 'located in the administrative territorial entity'],['C4', 'C6', 'country'], ['C5', 'C4', 'contains administrative territorial entity'], ['C5', 'C6', 'located in the administrative territorial entity'], ['C5', 'C6', 'country'], ['C6', 'C5', 'contains administrative territorial entity'], ['C7', 'C6', 'country of citizenship'], ['C8', 'C6', 'country of citizenship'], ['C14', 'C6', 'country of citizenship'], ['C15', 'C5', 'located in the administrative territorial entity'], ['C15', 'C6', 'country'], ['C19', 'C6', 'country of citizenship'], ['C1', 'C6', 'country'], ['C12', 'C6', 'country of citizenship'], ['C13', 'C6', 'country'], ['C9', 'C6', 'country']	

Table 1: Example of the DocRTE Task. Entity mentions of the same entity cluster are marked using same colors.

et al., 2023) adopted a span-based mention extractor and a table-filling approach for the entity clustering and relation classification sub-tasks. However, these classification and table-filling methods face issues with an excess of negative samples; for n identified entity mentions, there are $O(n^2)$ mention pairs to classify for clustering, most of which do not belong to the same cluster. A similar problem exists for relation classification, where relations are first identified at the entity mention pair level and then aggregated to the entity cluster pair level. This class imbalance issue is common in discriminative approaches for this task. In contrast, generative frameworks avoid this imbalance by design, as they inherently focus on extracting only positive samples—pairs that belong to the same cluster or share a relation — while ignoring negative samples.

As discussed, the single-stage generative approach does not address two key sub-tasks of DocRTE — mention extraction and entity clustering — while discriminative approaches face class imbalance issues due to their structural design. To overcome these challenges, we propose a novel three-stage generative framework, 3G-DocRTE, for DocRTE that effectively integrates both paradigms. In the first stage, we use a generative model to extract all entity mentions by linearizing mentions from the documents. In the second stage, we mark the identified mentions within the input documents and use a generative approach to normalize varying surface forms of the same entity into a unified entity cluster representation, with the first mention's surface form serving as the cluster representative. In the third stage, we employ the REBEL linearization scheme (Cabot and Navigli, 2021) to extract relational triples within the document. Our experiments on the DocRED dataset show that our approach outperforms previous generative models on all three sub-tasks of DocRTE and achieves competitive performance compared to SOTA discriminative models.

2 Task Formulation

Given a document D composed of L tokens, represented as $D = \{t_1, t_2, \ldots, t_L\}$, our objective is to perform document-level relation extraction. This task encompasses following three structured sub-tasks:

Entity Mention Extraction (EME): This task extracts all possible mention spans $M = \{m_i\}_{i=1}^{|M|}$ from the document, where each mention m_i is defined as a continuous sequence of tokens. Mathematically, a mention m_i is represented as $m_i = (t_s, t_{s+1}, \ldots, t_e)$, where $1 \leq s \leq e \leq L$ and $t_s, \ldots, t_e \in D$.

Entity Clustering and Typing (ECT): This task groups the extracted mentions into entity clusters and assigns an entity type, $E = \{(e_j, \tau_j)\}_{j=1}^{|E|}$. Mathematically, each cluster e_j is a set of mentions that are assumed to refer to the same real-world entity, i.e., $e_j = \{m_i | m_i \in M$ and m_i refers to entity $j\}$, and the type of each cluster is defined as $\tau_j \in \mathcal{T}$, where $\mathcal{T}$ is the set of all possible entity types.

Relational Triple Extraction (RTE): This task generates a set of relational triples $T = \{(e_j, r_{jk}, e_k) \mid e_j, e_k \in E, \; r_{jk} \in R \cup \{\perp\}\}$, where e_j and e_k are entity clusters, r_{jk} is selected from a predefined set $R \cup \{\perp\}$, with $\perp$ denoting the absence of any relation. The goal is to identify and specify the relations r_{jk} between each pair of entity clusters (e_j, e_k).

3 Proposed Framework: 3G-DocRTE

We introduce a three-step, multi-level generative framework, 3G-DocRTE, for Document-level relational triple Extraction (DocRTE), comprising (i) Entity Mention Extraction, (ii) Entity Clustering, and (iii) Relational Triple Extraction. First, we process documents containing multiple sentences to extract entity mentions. Next, we cluster these mentions to form entities along with their respective types. Finally, in the third stage, we generate relational triples present within the input document at the entity level.

A generative sequence-to-sequence model, such as BART (Lewis et al., 2019) models the probability of each output token o_i in the output sequence o based on the input sequence x and the previously generated output tokens $o_{<i}$: $\prod_{i=i}^{n} P(o_i \mid o_{<i}, x)$. The model is trained by maximizing the log-likelihood of the output tokens in the training data. We model this input and output sequence in an effective way for the three stages in our framework.

3.1 Entity Mention Extraction (EME)

Entity mentions can be extracted in a text either by using their specific tokens or by using their token index within the text. The same surface forms of an entity may appear multiple times throughout a document, making it difficult to ascertain precisely which unique instance is being referred to in token-based representation. Index-based representation of mentions can uniquely identify each occurrence. Given this advantage, we opt for an index-based approach to mention extraction in our framework.

We illustrate our index-based mention extraction strategy in Table 2 with an example. Each mention is identified by its start and end index position in the text. We append the start and end token index positions of all the mentions in a sequence separated by space. To maintain a consistent order during decoding, mentions are sorted according to their appearance in the input document. This enhances decoding efficiency by minimizing the token count. During decoding, we retain pairs of index positions, discarding single indexes if present at the end. Using these extracted start and end positions, we reconstruct the original surface form of each mention. To make index extraction easier for the pre-trained model, we follow Mallick et al. (2023) and insert the index of each token in the input document as well. Although, this increases the effective length of the document, but it helps the model during the mention generation.

Washington 0 Place 1 (2 William 3 Washington 4 House 5) 6 is 7 one 8 of 9 the 10 first 11 homes 12 built 13 by 14 freed 15 slaves 1 6 after 17 the 18 Emancipation 19 Proclamation 20 of 21 1863 22 in 23 Hampshire 24 County 25 , 26 West 27 Virginia 28 , 29 United 30 States 31 . 32 Washington 33 Place 34 was 35 built 36 by 37 William 38 and 39 Annie 40 Washington 41 in 42 north 43 Romney 44 between 45 1863 46 and 47 1874 48 on 49 land 50 given 51 to 52 Annie 53 by 54 her 55 former 56 owner 57 , 58 Susan 59 Blue 60 Parsons 61 of 62 Wappocomo 63 plantation 64 ...
0 1 3 5 19 20 22 22 24 25 27 28 30 31 33 34 38 38 40 41 44 44 46 46 48 48 53 53 59 61 63 64

Table 2: Example of input text and linearized output for mention extraction framework.

3.2 Entity Clustering & Typing (ECT)

To facilitate entity-level relational triple extraction, it is crucial to group local mentions of the same entity into document-level entity clusters, especially considering entities may have multiple mentions scattered throughout the input document and may exhibit various surface forms. Similar to our approach for mention extraction, we have introduced a linearization scheme tailored to enable entity clustering, also outputting cluster type information.

On the input side, we specify all mentions using start and end marker tags, denoted as <m> and </m>, respectively. For the output sequence of entity clustering framework, we use a linearization scheme where we replace each mention with the cluster-label/cluster-center. Additionally, we insert the entity type specific tags before and after each mention of that entity, as illustrated in Table 3. To simplify the decoding process and enhance efficiency, we opt to utilize the cluster center or cluster label rather than the entire cluster, thereby minimizing the number of tokens required. We decide to use the entity mention that appears first in the document for an entity cluster as the cluster-label/cluster-center.

For instance, In the example shown in Table 3, "<m> William Washington House </m>" is replaced by "<loc> Washington Place </loc>", where "Washington Place" serves as the first occurring

<m>Washington Place </m>(<m>William Washington House </m>) is one of the first homes built by freed slaves after the <m>Emancipation Proclamation </m>of <m>1863 </m>in <m>Hampshire County </m>, <m>West Virginia </m>, <m>United States </m>. <m>Washington Place </m>was built by <m>William </m>and <m>Annie Washington </m>in north <m>Romney </m>between <m>1863 </m>and <m>1874 </m>on land given to <m>Annie</m>by her former owner , <m>Susan Blue Parsons </m>of <m>Wappocomo plantation </m>...
<loc>Washington Place </loc>(<loc>Washington Place </loc>) is one of the first homes built by freed slaves after the <misc>Emancipation Proclamation </misc>of <time>1863 </time>in <loc>Hampshire County </loc>, <loc>West Virginia </loc>, <loc>United States </loc>. <loc>Washington Place </loc>was built by <per>William </per>and <per>Annie Washington </per>in north <loc>Romney </loc>between <time>1863 </time>and <time>1874 </time>on land given to <per>Annie </per>by her former owner , <per>Susan Blue Parsons </per>of <loc>Wappocomo plantation </loc>...

Table 3: Example of input and out representation for entity clustering stage.

<loc>Washington Place </loc>(<loc>Washington Place </loc>) is one of the first homes built by freed slaves after the <misc>Emancipation Proclamation </misc>of <time>1863 </time>in <loc>Hampshire County </loc>, <loc>West Virginia </loc>, <loc>United States </loc>. <loc>Washington Place </loc>was built by <per>William </per>and <per>Annie Washington </per>in north <loc>Romney </loc>between <time>1863 </time>and <time>1874 </time>on land given to <per>Annie </per>by her former owner , <per>Susan Blue Parsons </per>of <loc>Wappocomo plantation </loc>...
<triple>Washington Place <subj>United States <obj>country <triple>Emancipation Proclamation <subj>United States <obj>country <triple>Hampshire County <subj>West Virginia <obj>located in the administrative territorial entity <subj>United States <obj>country <triple>West Virginia <subj>Hampshire County <obj>contains administrative territorial entity <subj>United States <obj>located in the administrative territorial entity <subj>United States <obj>country <triple>United States <subj>West Virginia <obj>contains administrative territorial entity ...

Table 4: Example of input and output representation for RTE. Note that the blue-colored triple actually comprises two nested triples sharing the same subject/head entity.

mention for the cluster, with the entity type denoted as "<loc>".

During the decoding phase, each mention in the input document has a corresponding cluster label and a cluster type. We utilise these cluster labels to assign mentions to their respective clusters. Mentions sharing the same cluster label are grouped to form a cluster.

After this stage, the documents are normalized with respect to entity mentions as we replace the mentions with the corresponding cluster labels, and these are then enclosed within entity type marker tags. This normalization of the documents serves a dual purpose. Firstly, it simplifies the task of the subsequent entity triple extraction step. Secondly, it eliminates the need to output entire entity clusters with all their mentions, thereby effectively reducing the number of tokens required to be processed. This streamlined approach enhances both the efficiency and accuracy of the subsequent relational triple extraction stage.

3.3 Relational Triple Extraction (RTE)

In the final stage of our approach, namely relational triple Extraction, we focus on generating entity-level relational triples present within the documents. A relational triple comprises head and tail entities along with a relation from a predefined relation set. It's worth noting that a single document may express multiple relations between the same head and tail entities.

This stage builds upon the output of the previous entity clustering stage. To achieve effective linearization and denote all relational triples concisely, we adopt the linearization scheme proposed in the REBEL (Cabot and Navigli, 2021) paper as shown in Table 4. REBEL introduces a set of marker tokens for this purpose. Triples are grouped by the head entity, with the <triple> tag indicating the beginning of a new triple for a specific head entity, succeeded by the head entity itself. The <subj> tag marks the conclusion of the head entity, followed by the object entity. Subsequently, the <obj> tag signifies the conclusion of the tail entity and the initiation of the relation between the head and tail entities. In cases where there are multiple objects or relations of the same head entity, the <subj> tag marks the termination of the preceding relation, followed by the subsequent object entity. This process is repeated as needed for additional objects and relations. Once all relations involving a particular head entity have been processed, a fresh set of relations begins with the subsequent appearing head entity in the text. This iterative process continues until all triples have been linearized.

However, their proposed linearization scheme only utilizes the first occurring mention and disregards any remaining mentions of that entity. They extract mentions solely if they participate in one of the relational triples. Consequently, they do not comprehensively address the entity mention extraction and entity clustering sub-tasks.

To overcome this limitation, we opt to utilize the cluster label of the entities instead of solely relying on the first occurring mention of an entity. It's noteworthy that we have already extracted entity clusters along with entity types in the previous stage, and we can retrieve the entity cluster using the cluster label.

Each stage of our framework builds upon the results of the previous stage. By the conclusion of the third stage in our proposed framework, we can deduce all the relational triples present in the input document using the output of all three stages along with all entity mentions and entity clusters in the documents. In this way our proposed three-stage generative framework 3G-DocRTE solves all three sub-tasks of document-level relational triple extraction.

4 Experiments

4.1 Dataset & Evaluation Metric

We conduct our experiments using the manually annotated part of DocRED dataset (Yao et al., 2019) and use the splits provided by JEREX (Eberts and Ulges, 2021). JEREX removed 45 erroneous document from training set, used 3,008 documents for the training. They randomly split the 1,000 documents in the original dev set into two parts: 300 documents as validation set, and 700 documents for the test set. The specific statistics for these JEREX splits are detailed in Table 5.

Split	#Doc	#Men	#Ent	#Rel
Train	3,008	78,677	58,708	37,486
Dev	300	7,702	5,805	3,678
Test	700	17,988	13,594	8,787

Table 5: DocRED dataset split used for DocRTE.

As the dataset split, we use the evaluation methodology of Eberts and Ulges (2021) for this task. We adopt a strict evaluation criteria for all three sub-tasks of DocRTE. An **entity mention** is considered correct if its surface form is an exact match with a ground truth mention's surface form. An **entity cluster** is considered correct only if it exactly matches a ground truth entity cluster. This includes — all mentions within the generated cluster matching exactly with those in a ground truth cluster — and the cluster type also matching with the ground truth type. We generate relational triples at entity-cluster-level. A **relational triple** is considered as correct if the head and tail entities, as well as the relation itself, are correct and matches with the ground truth relational triple. For each of these sub-tasks, we report precision, recall, and F1 scores.

4.2 Baselines

For baselines, we use two generative approaches: REBEL (Cabot and Navigli, 2021) and Seq2Rel (Giorgi et al., 2022), two discriminative approaches: JEREX (Eberts and Ulges, 2021) and TAG (Zhang et al., 2023) for comparison.

REBEL (Cabot and Navigli, 2021): REBEL uses a BART (Lewis et al., 2019) model to generate the relational triples in a sequence-to-sequence fashion. They use a linearization scheme where entities and relations are represented as tokens separated by special tags. However, this approach cannot solve the mention extraction and entity clustering sub-tasks for the DocRTE.

Seq2Rel (Giorgi et al., 2022): This is another Seq2Seq approach where they use BERT encoder and LSTM decoder to generate the relational triples using a pre-defined linearization scheme. They designed the linearization scheme in such a way that it can extract the entity clusters along with the triples. But this approach only includes those entity mentions and entity clusters that participate in some relational triples. Entity mentions and clusters that are not part of any relational triples are ignored in their linearization scheme. Also, they extract an entity cluster as many times as they participate in as many triples. Additionally, there is a significant amount of redundant entity cluster generation in this approach.

JEREX (Eberts and Ulges, 2021): This is a 3-step discriminative approach for the DocRTE task. First, they extract the entity mentioned using a span-based classifier. Next, they classify each pair of extracted mentions if they belong to the same entity cluster or not. In the third step, they classify the relations or no relation among all possible pairs of entity clusters. They can train these three stages either in a pipeline fashion or in a joint fashion.

TAG (Zhang et al., 2023): This model proposed a table-filling approach for DocRTE. First, it identifies the entity mention spans and creates a table where rows and columns of the table represent each mention. Each cell of this table is then filled with values that represent if they belong to the same cluster or not and the relations between the mentions. Some aggregation mechanism is used to obtain the entity cluster pair-level relations from the mention pair level relations.

4.3 Parameter Settings

For training, we mostly follow the REBEL paper (Cabot and Navigli, 2021). We use BART-large (Lewis et al., 2019) as our base model and fine-tune it separately on the DocRED human annotated dataset for each sub-task of DocRTE with sub-task specific linearization schemes. We used batch size of 4 and AdamW (Loshchilov and Hutter, 2019) optimizer with learning rate at 1e-05, the weight decay at 1e-03. Additionally, The REBEL paper (Cabot and Navigli, 2021) released a pre-trained version of BART-large, which was fine-tuned on a relational triple dataset derived from Wikipedia hyperlinks. We also utilize this pre-trained BART-large model for our experiments, referring to it with the '-pt' suffix.

5 Experimental Results

The performance comparison of generative models and discriminative models are summarized in Table 6.

Entity Mention Extraction: In the entity mention extraction sub-task, both the 3G-DocRTE and 3G-DocRTE-pt models demonstrate competitive performance, each achieving an F1-score of 0.930, closely matching other baseline models. This performance indicates that our 3G-DocRTE framework effectively identifies correct mention spans across various document contexts.

Entity Clustering & Typing: In entity clustering & typing, the 3G-DocRTE framework shows strong performance with an F1-score of more than 80%. This represents an almost 30% higher F1 score than that achieved by the REBEL framework for this task. When type information is not considered in evaluation like TAG does, our approach achieves competitive performance. These results highlight the robustness of our framework in effectively grouping mentions into accurate clusters. However, our performance is slightly lagging—about 5% in F1 score—behind the best result in EC.

Relational Triple Extraction: In relational triple extraction, generative models generally exhibit lower F1-scores compared to discriminative models as evident from Table 6. 3G-DocRTE-pt records the highest F1-score among generative models at 0.405 under the strict evaluation criterion. Under the relaxed criterion, TAG model achieves the highest F1 score of 43.2%, whereas our approach achieves around 41.2% F1 score. The general performance gap between the discriminative and generative models underscores the challenges and potential trade-offs inherent in generative approaches, highlighting a critical area for further improvement. Particularly, our generative approach achieves significantly lower performance in the entity clustering task which needs more attention in future.

Overall, while discriminative models tend to show slight advantages in specific sub-tasks, particularly in Entity Clustering, our 3G-DocRTE framework, particularly in its pre-trained variant, consistently delivers competitive and balanced performance across all sub-tasks. The consistent performance of our framework underscores its potential to advance the state-of-the-art in document-level relation extraction, highlighting its capability to handle complex relational data effectively.

6 Analysis & Discussion

6.1 Discriminative vs Generative Performance

From Table 6, it is evident that discriminative models generally outperform generative models. This discrepancy can likely be attributed to the inherent design choices between these paradigms, which affect the volume of effective training samples. Discriminative models train on all possible pairs of entity clusters to identify relations, using a larger number of training samples per document. During inference, they identify relations from these pairs and aggregate these into document-level triples. In contrast, generative models are trained directly on documents; a single document outputs a set of triples. Considering the DocRED training data, which comprises approximately 3,000 documents with about 58,000 entity clusters, the effective training sample size for discriminative models significantly exceeds that of generative models. This considerable difference may be a reason for the better performance observed in discriminative models.

6.2 Copy vs Reasoning in 3G-DocRTE

We analyze how generative frameworks perform in tasks that involve only copying versus those requiring some reasoning. In the case of the Marker-Inserted linearization scheme (see Table 7) for entity mention extraction, our model simply needs to copy the input tokens and insert <m> tags where entity mentions occur. Identifying a mention is

	EME			ECT			RTE		
Model	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
REBEL	0.844	0.444	0.582	0.727	0.367	0.488	0.237	0.223	0.230
REBEL-pt	0.837	0.449	0.584	0.720	0.362	0.482	0.251	0.249	0.250
Seq2Rel	-	-	-	-	-	-	0.440	0.338	0.382
JEREX	0.933	0.927	0.930	0.798	0.804	0.801	0.428	0.383	0.404
TAG	0.929	0.928	0.929	0.811	0.798	0.804	0.428	0.395	0.411
3G-DocRTE	0.933	0.926	0.930	0.810	0.807	0.808	0.385	0.376	0.381
3G-DocRTE-pt	0.930	0.930	0.930	0.802	0.805	0.804	0.413	0.397	0.405

Table 6: Performance comparison of generative/discriminative models against 3G-DocRTE framework on JEREX split of DocRED. Models marked with '-pt' denote a BART-large model variant that is post-trained using the REBEL dataset.

a localized task that does not require reasoning across documents. Hence, in this task, our framework achieves a very high F1 score of around 93%. Contrarily, for the entity clustering task, the linearization scheme used in Table 3, our generative framework must not only copy most tokens from the input text but also resolve co-references among different mentions. This co-reference resolution involves long-term reasoning across the entire document. As shown in Table 6, our model achieves an F1 score of approximately 80% in the entity clustering task, which is 10% lower in terms of absolute F1 score compared to the mention extraction task. This performance difference between the two tasks indicates that auto-regressive generative models struggle with reasoning tasks while decoding the output sequence.

6.3 Ablation for Entity Mention Extraction

To optimize mention extraction strategies within the 3G-DocRTE model, we conducted ablation studies focusing on different linearization schemes. Apart from the index-based scheme discussed in Section 4.1, we explored both the marker-inserted and marker-separated schemes. In the marker-inserted scheme, the start and end of all the entity mentions in the document are marked by <m> and </m> tags. An example of this approach can be seen in row 2 of Table 7. The marker-separated scheme includes only the entity mention tokens in the output sequence, marking the start of each mention with a <m> tag. An example of this can be found in row 3 of Table 7. The performances of these schemes are reported in Table 8, showing comparable results. Additionally, we evaluated an adaptation of the index-based scheme that omits the token index in the input document. This approach (see Table 12 in Appendix) resulted in a significant drop in the F1 score for the mention extraction task, as shown in row 4 of Table 8). We use the Index Based linearization for the final model as it achieves high F1 score with fewer output tokens.

Washington Place (William Washington House) is one of the first homes built by freed slaves after the Emancipation Proclamation of 1863 in Hampshire County , West Virginia , United States ...
Marker-Inserted: <m>Washington Place </m>(<m>William Washington House </m>) is one of the first homes built by freed slaves after the <m>Emancipation Proclamation </m>of <m>1863 </m>in ...
Marker-Separated: <m>Washington Place <m>William Washington House <m>Emancipation Proclamation <m>1863 ...

Table 7: Example of token-based linearization strategy for Mention Extraction using start (<m>) and end (</m>) marker tags.

	EME		
Linearization Scheme	**P**	**R**	**F1**
Marker-Inserted	0.939	0.929	0.934
Marker-Separated	0.925	0.924	0.925
Index Based	0.930	0.930	0.930
- w/o index in document	0.532	0.527	0.529

Table 8: Performance comparison of two linearization scheme for entity mention extraction.

6.4 Ablation for Entity Clustering & Typing

In addition to the entity clustering linearization scheme described in Section 4.2 (referred to as Type-Marker-Inserted), we experiment with another scheme, referred to as Type-Marker-Separated, similar to the previous scheme. For every tagged mention in input text, we extract an entity type marker tag followed by the cluster label for that mention. We choose the first appearing mention of a cluster as a cluster label. The performance comparison of these two linearization schemes is reported in Table 10. Results indicate that the Type-Marker-Inserted scheme slightly outperforms the Type-Marker-Separated scheme. The results of our evaluation indicate that the Type-Marker-Inserted scheme slightly outperforms the Type-Marker-Separated scheme. This performance difference suggests that the Type-Marker-Inserted approach forms a coherent and meaningful text compared to the disconnected and divided format of the Type-Marker-Separated scheme which may facilitate better understanding by the model and enables effective grouping of the mentions.

Type-Marker-Separated: <loc>Washington Place <loc>Washington Place <misc>Emancipation Proclamation <time>1863 <loc>Hampshire County <loc>West Virginia <loc>United States <loc>Washington Place <per>William <per>Annie Washington <loc>Romney <time>1863 <time>1874 <per>Annie <per>Susan Blue Parsons <loc>Wappocomo plantation ...

Table 9: Type-Marker-Separated scheme for ECT using the same input format as described in Table 3.

	ECT		
Linearization Scheme	**P**	**R**	**F1**
Type-Marker-Separated	0.792	0.796	0.794
Type-Marker-Inserted	0.802	0.805	0.804

Table 10: Performance comparison of two different linearization schemes for ECT task.

6.5 Ablation for Relational Triple Extraction

Before REBEL Cabot and Navigli (2021), Nayak and Ng (2020) proposed another linearization scheme with their Word Decoder model where each triple is separated by a special tag and the components of a triples (head entity, tail entity, and a relation) are separated by another special tag (see Table 13 in Appendix for more details). We experimented with such representation for the relational triple extraction stage of 3G-DocRTE and include the results in Table 11. The evaluation shows that the REBEL representation yields better performance compared to the Word Decoder representation. It is compact and requires significantly fewer tokens to represent all the relational triples in a document.

	RTE		
Linearization Scheme	**P**	**R**	**F1**
Word Decoder	0.363	0.395	0.378
REBEL	0.413	0.397	0.405

Table 11: Performance comparison of RTE task with two different output representations.

6.6 Unified EME and ECT Approach

As document-level tasks involve a large number of tokens, maximum token length often becomes a performance bottleneck for any pre-trained model such as BART. Implementing all three steps—mention extraction, entity clustering, and relation extraction—within a single linearization process can increase token length and lead to unnecessary repetition of clusters. So we propose three stages for three sub-tasks of DocRTE.

But **Is it possible to reduce the number of steps in the pipeline?** Of the three stages in our proposed 3G-DocRTE framework, it appears feasible to combine the first two stages—mention extraction and entity clustering—into a single step and use a single generative model which takes plain text as input and generates output similar to the output of the second stage of 3G-DocRTE. Our goal is to replace each mention of an entity in the documents with corresponding cluster labels enclosed by entity-type markers. Although the generative approach can perform this combined task, but a significant challenge was to map these cluster labels back to the original mentions in documents as cluster labels and their mentions are not always of the same token length. The BART tokenizer alters the text by removing what it perceives as extra spaces, and it can split tokens into sub-tokens or merge them, complicating the recovery of the original tokens. So we believe that it is more effective and intuitive to use two different stages of the generative approach for these two sub-tasks of mention extraction and entity clustering in DocRTE.

7 Related Work

Relational Triple extraction (RTE) is a crucial task for extracting knowledge from text, where this knowledge is represented in triple form, consisting of two entities (subject and object) and a directed relation from the subject to the object. These triples can be added to knowledge bases (KBs) to enrich them. There are two distinct approaches to addressing this task: (i) Relation Classification (RC) and (ii) Relational Triple Extraction (RTE). In the relation classification approach, entities are pre-identified, and models are required to identify the relations, or 'no relation', between pairs of entities. In the relational triple Extraction approach, models simultaneously extract corresponding entity pairs and their relations. Recently, RTE approaches have gained popularity as they provide an end-to-end solution for this task.

Mintz et al. (2009) introduced the distant supervision method to generate large-scale datasets for relation Classification task without the need for human annotations. It has significantly fostered the research in this area. Following the introduction of word embeddings in NLP (Mikolov et al., 2013; Pennington et al., 2014), numerous neural models were proposed to address this task. Zeng et al. (2014, 2015) introduced CNN-based models for classifying relations or 'no relation' between two entities within a short sentence-level context. Additionally, Jat et al. (2018); Nayak and Ng (2019) proposed attention models for the same task.

Relational triple extraction is relatively new task and Zheng et al. (2017) was very first to introduce a tagging-based approach for this task, while Zeng et al. (2018); Nayak and Ng (2020) explored sequence-to-sequence learning for the same task. Eberts and Ulges (2021) used a pre-trained BART based model that represents relational triples using a linearization mechanism. The BART decoder of their model can generate this representation in an auto-regressive manner. Recent models have leveraged pre-trained transformers like BERT (Devlin et al., 2019) to encode the sentences to get a better representation. Models such as TPLinker (Wang et al., 2020b), 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) have proposed various neural architectures based on BERT to address RTE at the sentence level.

Recently, with the introduction of the DocRED dataset (Yao et al., 2019), document-level relation extraction has gained significant traction in the research community. Initially, in this field, most research work focused on relation classification at the entity levels within documents. Nan et al. (2020); Wang et al. (2020a); Zeng et al. (2020, 2021); Xu et al. (2021b,a) introduced various attention models and graph convolution models for this task. More recently, researchers have explored the document-level relational triple extraction task on the DocRED dataset. This task is notably more challenging than its sentence-level counterpart, as it involves longer context lengths, requires coreference resolution across the longer text, and the models need to perform multi-hop reasoning to extract triples. REBEL (Cabot and Navigli, 2021) and Seq2Rel (Giorgi et al., 2022) have proposed sequence-to-sequence models for document-level relational triple extraction using a linearized triple representation. Alternatively, JEREX (Eberts and Ulges, 2021), Joint-M (Xu and Choi, 2022), and TAG (Zhang et al., 2023) have proposed multi-stage discriminative approaches for the same task.

8 Conclusion

In this work, we propose an effective way of using generative frameworks for the document-level relational triple extraction task (DocRTE). Our approach completely addresses all three sub-tasks of DocRTE: entity mention extraction, entity clustering, and relational triple extraction, whereas, the previous generative approaches exhibit significant deficiencies in managing these sub-tasks. On the DocRED dataset, our proposed framework surpasses earlier generative models. Additionally, when compared with multi-stage discriminative approaches on the same dataset, our method achieves competitive performance across the three sub-tasks of DocRTE.

9 Ethics Statement

There is no ethical issues concerning this research work.

10 Limitations

Due to limited GPU availability, we can only fine-tune the BART model at this time. To achieve broader applicability, fine-tuning other encoder-decoder models like T5 would be advantageous.

Another potential limitation of our approach is the significant increase in document length when additional tokens are inserted. While this does not pose an issue for the DocRED dataset, it could become problematic for longer documents. Specifically, the additional tokens may exceed the maximum token limit allowed in the encoder.

References

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. Rebel: Relation extraction by end-to-end language generation. In *EMNLP*.

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

Markus Eberts and Adrian Ulges. 2021. An end-to-end model for entity-level relation extraction using multi-instance learning. In *Conference of the European Chapter of the Association for Computational Linguistics*.

John Giorgi, Gary D Bader, and Bo Wang. 2022. A sequence-to-sequence approach for document-level relation extraction. In *Workshop on Biomedical Natural Language Processing*.

Sharmistha Jat, Siddhesh Khandelwal, and Partha Pratim Talukdar. 2018. Improving distantly supervised relation extraction using word and entity based attention. *AKBC*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdel rahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*.

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*.

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

Prabir Mallick, Tapas Nayak, and Indrajit Bhattacharya. 2023. Adapting pre-trained generative models for extractive question answering. In *GEM Workshop at EMNLP*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Tapas Nayak and Hwee Tou Ng. 2019. Effective attention modeling for neural relation extraction. In *Proceedings of the Conference on Computational Natural Language Learning*.

Tapas Nayak and Hwee Tou Ng. 2020. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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*.

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

Difeng Wang, Wei Hu, Ermei Cao, and Weijian Sun. 2020a. Global-to-local neural networks for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020b. 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*.

Liyan Xu and Jinho Choi. 2022. Modeling task interactions in document-level joint entity and relation extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5409–5416, Seattle, United States. Association for Computational Linguistics.

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021a. Discriminative reasoning for document-level relation extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021b. Document-level relation extraction with reconstruction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14167–14175.

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

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 ACL 2019*.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the 25th International Conference on Computational Linguistics*.

Shuang Zeng, Yuting Wu, and Baobao Chang. 2021. SIRE: Separate intra- and inter-sentential reasoning for document-level relation extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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*.

Ruoyu Zhang, Yanzeng Li, and Lei Zou. 2023. A novel table-to-graph generation approach for document-level joint entity and relation extraction. In *Annual Meeting of the Association for Computational Linguistics*.

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 *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

A Appendix

A.1 Examples of Linearization

We include some examples of linearization schemes used in our ablation studies here. Details are included in the respective caption.

Washington Place (William Washington House) is one of the first homes built by freed slaves after the Emancipation Proclamation of 1863 in Hampshire County , West Virginia , United States . Washington Place was built by William and Annie Washington in north Romney between 1863 and 1874 on land given to Annie by her former owner , Susan Blue Parsons of Wappocomo plantation ...
0 1 3 5 19 20 22 22 24 25 27 28 30 31 33 34 38 38 40 41 44 44 46 46 48 48 53 53 59 61 63 64

Table 12: Example of input text and linearized output for mention extraction framework where we do not insert the token index in the input documents.

<loc>Washington Place </loc>(<loc>Washington Place </loc>) is one of the first homes built by freed slaves after the <misc>Emancipation Proclamation </misc>of <time>1863 </time>in <loc>Hampshire County </loc>, <loc>West Virginia </loc>, <loc>United States </loc>. <loc>Washington Place </loc>was built by <per>William </per>and <per>Annie Washington </per>in north <loc>Romney </loc>between <time>1863 </time>and <time>1874 </time>on land given to <per>Annie </per>by her former owner , <per>Susan Blue Parsons </per>of <loc>Wappocomo plantation </loc>...
<triple>Washington Place <subj>United States <obj>country <triple>Emancipation Proclamation <subj>United States <obj>country <triple>Hampshire County <subj>West Virginia <obj>located in the administrative territorial entity <subj>United States <obj>country <triple>West Virginia <subj>Hampshire County <obj>contains administrative territorial entity <subj>United States <obj>located in the administrative territorial entity <subj>United States <obj>country <triple>United States <subj>West Virginia <obj>contains administrative territorial entity ...
<triple>Washington Place <subj>United States <obj:>country <triple>... <triple>Hampshire County <subj:>West Virginia <obj:>located in the administrative territorial entity <triple>Hampshire County <subj:>United States <obj:>country <triple>...

Table 13: Example of input and output representation for RTE for REBEL and Word Decoder models. First row represents the input document format used for both of these two models. Row 2 represents the REBEL (Cabot and Navigli, 2021) representation for output triples. Row 3 shows the Word Decoder (Nayak and Ng, 2020) representation for triples. In REBEL representation if you look at the blue-colored triple, you see that two triples are nested which share the same subject/head entity. But the same two triples in Word Decoder representation are flattened for simpler representation.