REXEL: An End-to-end Model for Document-Level Relation Extraction and Entity Linking

Nacime Bouziani*1, Shubhi Tyagi2, Joseph Fisher2, Jens Lehmann2, Andrea Pierleoni2

1 I-X Centre for AI In Science
Imperial College London, London, UK
2 Amazon Alexa AI
Cambridge, UK

n.bouziani18@imperial.ac.uk
{tshubhi, fshjos, jlehmnn, apierleo}@amazon.com

Abstract

Extracting structured information from unstructured text is critical for many downstream NLP applications and is traditionally achieved by closed information extraction (cIE). However, existing approaches for cIE suffer from two limitations: (i) they are often pipelines which makes them prone to error propagation, and/or (ii) they are restricted to sentence level which prevents them from capturing long-range dependencies and results in expensive inference time. We address these limitations by proposing REXEL, a highly efficient and accurate model for the joint task of document level cIE (DocIE). REXEL performs mention detection, entity typing, entity disambiguation, coreference resolution and document-level relation classification in a single forward pass to yield facts fully linked to a reference knowledge graph. It is on average 11 times faster than competitive existing approaches in a similar setting and performs competitively both when optimised for any of the individual subtask and a variety of combinations of different joint tasks, surpassing the baselines by an average of more than 6 F1 points. The combination of speed and accuracy makes REXEL an accurate cost-efficient system for extracting structured information at web-scale. We also release an extension of the DocRED dataset to enable benchmarking of future work on DocIE, which will be available at https://github.com/amazon-science/e2e-docie.

1 Introduction

Extracting structured information from unstructured text is a critical step for many downstream NLP tasks like knowledge graph construction (Muhammad et al., 2020), question answering (Yao and Van Durme, 2014), knowledge discovery (Trisedya et al., 2019), and text summarization (Genest and Lapalme, 2012). In cIE, this is defined as extracting an exhaustive set of (subject, relation, object) triples, or facts, from unstructured text that are fully linked, i.e., consistent with a predefined set of entities and relations from a knowledge graph (KG) schema. cIE can be further decomposed into the subtasks: mention detection (MD), entity typing (ET), entity disambiguation (ED), and relation classification (RC).

Traditionally, cIE is done by combining these subtasks sequentially (Nasar et al., 2021), which involves the use of separate and often different models for each task to yield facts that can be ingested into a KG. However, such pipeline architectures are prone to error accumulation from each component leading to significant deterioration of the overall performance (Miwa and Sasaki, 2014; Trisedya et al., 2019; Mesquita et al., 2019). Additionally, pipeline architectures assume a one-way dependency between the subtasks, disregarding the dependencies among components that could effectively boost performance. For instance, while ED typically informs RC, recent works have demonstrated that RC information can also be effectively utilised for the ED task (Ayoola et al., 2022a), and help preventing issues such as popular entities overshadowing less common entities (Provatorova et al., 2021). Consequently, various joint/end-to-end (E2E) systems have been proposed to address this issue by jointly performing NER and RC (Miwa and Sasaki, 2014; Pawar et al., 2017). This joint task is often referred to as relation extraction (RE). However, these approaches do not address ED and thus do not yield facts fully linked to a KG.

Another drawback of existing approaches for cIE is that they mostly operate at sentence level, i.e., perform RC between two entities from a single sentence at a time (Cai et al., 2016; Han et al., 2018; Feng et al., 2018). Thus, they capture limited
sentence-level context and miss the facts that are expressed between entities across sentences. This severely limits the amount of information that can be extracted from the web. According to (Yao et al., 2019), 40.7% of the facts in a document can only be determined at the document level. Also, sentence-level approaches require a forward pass for each sentence, often leading to higher inference times, which makes them inefficient for web-scale applications. In contrast, document-level RC is computationally more efficient as it extracts triples over an entire document in a single forward pass. To address these issues, several models have been proposed for document-level RC (Zeng et al., 2020; Wang et al., 2020; Xu et al., 2021; Zhang et al., 2021) but they do not perform the remaining sub-tasks needed for DocIE.

To address the above problems we introduce REXEL, a computationally efficient E2E model for DocIE. REXEL takes unstructured text and extracts facts which are fully linked to a reference KG in a single forward pass per document. It has a modular architecture in which the various subtasks for DocIE inform each other by leveraging intermediate embedding representations. Thus, the proposed framework facilitates deployment not only for DocIE but also for various combinations of its 5 subtasks (e.g., use MD and ET only for NER). The combination of modularity, fast inference speed and high accuracy makes REXEL suitable for performing DocIE or its sub-tasks at industry scale.

To summarize, our contributions are as follows:

1. We introduce REXEL, a unified E2E model for DocIE, i.e., extracting facts at document level fully linked to a reference KG in a single forward pass per document.

2. We demonstrate that though REXEL is optimised for the E2E task of DocIE, it maintains a competitive edge with related work in E2E RE setting and all its individual subtasks. Specifically, REXEL improves upon the baselines for the E2E RE task by an average of >6 F1 points across datasets. When comparing the performance of individual subtasks, we observe that REXEL outperforms the baselines by an average of 6 F1 points.

3. We also demonstrate that when compared to other E2E RE models, in the same setting REXEL is on average 11 times faster.

4. Finally, we release an extension of the DocRED (Yao et al., 2019) dataset released by (Eberts and Ulges, 2021) augmented with silver standard labels for entity linking to facilitate benchmarking of future work on DocIE. We name this extension DocRED-IE.

2 Related work

2.1 Closed Information Extraction (cIE)

Several E2E systems have been proposed for cIE (Liu et al., 2018; Trisedya et al., 2019; Sui et al., 2021; Josifoski et al., 2022). However, all these...
methods are sentence-level architectures and therefore they inherently lose triples expressed across sentences. They are also prohibitively expensive for deployment at web-scale since the inference compute increases linearly with the number of sentences requiring a forward pass for each sentence.

In comparison, DocIE is a significantly more challenging task as it involves capturing long-range dependencies effectively to extract relations between entities which are further apart from each other in the text. Scaling cIE to document level from sentence level also requires an additional sub-task of coreference resolution (Coref), i.e., group all the different mentions in the document referring to the same entity.

### 2.2 Document-level Relation Extraction

Various E2E models have been proposed that combine the task of NER and document-level RC in a joint setting (Eberts and Ulges, 2021; Zaporojets et al., 2021). Other works such as REBEL (Huguet Cabot andNavigli, 2021) and KBIE (Verlinden et al., 2021) have proposed using additional data like the Wikipedia text, hyperlinks and Wikidata KG to further improve RE performance. However, these approaches do not perform ED and hence do not yield facts fully linked to a reference KG. Thus, ingesting the output of such models in a KG necessitates a separate ED model to link the extracted entities. This again results in a pipeline architecture between RE and ED models.

To the best of our knowledge, REXEL is the first E2E model to extract facts which are fully linked to a reference KG, at document level and address the task of DocIE. Also, while relation classification (RC) is also usually referred to as relation extraction (RE), the E2E literature has adopted different conventions. For sake of consistency with prior works (Eberts and Ulges, 2021; Miwa and Bansal, 2016), we use RC to refer to the extraction of relations between entity pairs and RE to refer to the E2E task including MD, ET, RC, and Coref.

### 3 REXEL

We introduce REXEL (Relation Extraction and Entity Linking), a novel end-to-end model for DocIE. REXEL extracts triples fully linked to a KG by jointly performing MD, ET, document level RC, Coref and ED in a single forward pass. It combines the 5 subtasks in a unified architecture via intermediate embedding representations. This facilitates each task to inherently benefit from each other, significantly boosting task accuracy, extracting facts expressed across sentences, and maintaining computational efficiency for web-scale deployment. Figure 1 illustrates the architecture and each module is detailed in following sections.

#### 3.1 Task Formulation

Given a KG with a set of entities $E = \{e_1, e_2, \ldots, e_{|E|}\}$, entity types $T = \{t_1, t_2, \ldots, t_{|T|}\}$, and relations $R = \{r_1, r_2, \ldots, r_{|R|}\}$, let $X = \{x_1, x_2, \ldots, x_{|X|}\}$ be the sequence of tokens in a document (d). The goal of DocIE is to extract linked facts, i.e., $G : X \rightarrow G$ with $G \subseteq E \times R \times E$ being a set of triples. This is done by (i) MD: extracting mention spans resulting in a list of subsets of $X$, (ii) Coref: clustering mentions into entities, (iii) ET: extracting the entity types for each cluster, (iv) RC: extracting relations by mapping entity pairs $\{e_1, e_2\}$ to relations $r \in R$ and (v) ED: assigning each cluster of mentions to a corresponding KG entity $e \in E$.

#### 3.2 Mention Detection (MD)

We encode the tokens $x_i$ in the input text document using RoBERTa (Liu et al., 2019) and use the contextualised token embeddings $h_i$ from the final layer of the encoder for the token $x_i$. The tokens are encoded using the BIO tagging format (Ramshaw and Marcus, 1995). We then train a linear layer to perform token classification from the token embeddings $h_i$ using cross-entropy loss $L_m$ with respect to the gold token labels. We obtain mention embeddings $m_i$ for each mention $m_i$ by average pooling the contextualised token embeddings ($h_i$) for all tokens in a mention from the final transformer layer. The output of this module is a list of mention spans present in the input text along with their contextualised embeddings.

#### 3.3 Entity Typing (ET)

Given a fixed set of types $t \in T$, the ET module is trained by applying a linear layer $f_1$ followed by a sigmoid activation to the mention embedding $m_i$ to predict an independent unnormalised score for each type $t$ for each mention $m_i$. REXEL produces two independent predictions for ET. The $ET_{ed}$ layer predicts fine-grained Wikidata types (1.3k) that are later used to inform ED. We do not train on this explicitly, but via ED (see Section 3.6). The $ET_{final}$ layer predicts the type(s) for each mention according to the ones permissible within the target dataset for the target task. We train this module from the
gold entity types using binary cross-entropy loss $L_t$ corresponding to $ET_{final}$ predictions. There are two separate predictions for ET as the target dataset may not have as many or the same fine-grained types. Fine-grained entity types provide critical additional information that can inform ED and thus boost overall performance. We aggregate predictions at entity cluster level by selecting the most frequent type among the cluster mentions as the entity type. REXEL supports both single and multiple type classification.

3.4 Relation Classification (RC)
REXEL extracts relations at mention-level using a cross-attention transformer and uses the coreference resolution predictions to map the extracted relations to the entity-level. We employ top-k pruning from (Lee et al., 2018) to extract relations only for the $k$ mention pairs with highest probabilities of being connected by a relation. This probability is computed for each mention pair using a bilinear layer. This first stage results in less accurate but more efficient predictions and is referred to as the coarse stage. However, in REXEL the coarse stage is adopted for both: relation classification and coreference resolution. The coarse stage is then followed by the fine stage, which extracts relations between surviving mention pairs. The resulting coarse-to-fine RC module yields competitive accuracy with high efficiency. Similar to the ET module, we have multiple prediction layers for RC: $RC_{ed}$, which predicts the Wikidata relations and is used as an input to the ED module, $RC_{coref}$, which predicts the pairwise coreference scores for the Coref module, and $RC_{final}$, which is the final prediction layer on the target relations of the given dataset. This module is trained from the gold mention spans, gold entity types, gold entity IDs and gold clusters using binary cross-entropy loss $L_c$ with respect to the gold triples on the $RC_{final}$ prediction layer only.

3.5 Coreference Resolution (Coref)
This module has two stages: the first predicts pairwise coreference scores for each mention pair that remains after top-k pruning, and the second uses pair predictions to form entity clusters by using average linkage clustering based on a given distance threshold. Other approaches like greedy clustering, complete linkage and clustering via Wikidata identifiers resulted in similar performance. More details can be found in Appendix A. The first stage can be expressed as a relation classification task with one relation that determines whether two mentions are coreferent to each other. Hence, we delegate this stage to the RC cross-attention transformer. The training is done with respect to the predicted coreference scores only. We train this module from the pairwise scores of the gold mention spans using binary cross-entropy loss $L_d$ with respect to the gold clusters. The output of this module is a group of entity clusters in a document and their corresponding mentions.

3.6 Entity Disambiguation (ED)
REXEL links each entity mention in the text to a unique Wikidata ID using a training procedure similar to (Ayoola et al., 2022a). The ED module takes as input the mention embeddings $m_i$, entity type predictions for ED $t_{ed}$ and RC predictions for ED $r_{ed}$. We also add a global entity prior $P(e|m)$ (PEM score), which is the probability of an entity given the mention text and is obtained from hyperlink count statistics as done in (Raiman and Raiman, 2018). We train this module from gold mention spans and gold entity types $t_{ed}$ by using binary cross-entropy loss $L_d$ with respect to the gold entity IDs. Note that we do not train on $ET_{ed}$ and $RC_{ed}$ explicitly, instead, the training for those predictions is done using the signal from $L_d$ only. REXEL performs ED for each mention and we get the entity IDs at the cluster level (i.e., when multiple mentions are clustered together by coref) by taking the majority vote of the entity IDs for all the mentions in the cluster.

3.7 Optimization and Inference
REXEL is optimised using a weighted sum of the module-specific losses with fixed weights, which are tunable hyperparameters as follows:

$$
L = \lambda_1 L_m + \lambda_2 L_t + \lambda_3 L_d + \lambda_4 L_c + \lambda_5 L_r \quad (1)
$$

When training on a single subtask, the weights for all the other task losses are set to zero. When training for the RE task, $\lambda_3$ is set to zero. For individual subtask inference, we use gold labels for the other tasks. For the RE inference, we use the predicted mention spans, predicted entity types, predicted coref clusters and predicted entities as input. Training environment details are in Appendix B.

4 Experiments
4.1 Datasets
We report performance on DWIE (Zaporojets et al., 2021), the only dataset available supporting Do-
cIE. We also augment the end-to-end DocRED split (DocRED-E2E) (Eberts and Ulges, 2021), which does not support annotations for ED, with silver annotations for entity links, and release the resulting dataset for future works. For this, we use the SoTA EL model ReFinED (Ayoola et al., 2022b) to link the mention spans against Wikidata and report DocIE performance on the DocRed-E2E split augmented with these entity links. We also report performance on DocRED-E2E for the E2E RE task, which allows comparison with existing approaches. More details on the datasets can be found in Appendix C.

4.2 Evaluation settings

4.2.1 Subtask

In the **Subtask** training setting, we train and evaluate each of the 5 DocIE subtasks independently as mentioned in 3.7. This setting measures the ceiling performance of each component. We report these metrics to understand the impact of the performance of each component as we move from independent subtask training to E2E RE and E2E DocIE training settings.

4.2.2 Relation Extraction (RE)

Despite the recent works on the joint entity and relation extraction task for document-level RE, there has been a lack of a cohesive task definition and consistent baselines, leading to discrepancies in dataset usage and evaluation procedures, as discussed in (Taillé et al., 2021). We follow the **hard-metric** setting to evaluate the E2E RE task in line with previous works (Eberts and Ulges, 2021; Zaporojets et al., 2021). More precisely, a triple is considered as correct if the relation type and the entity clusters associated to the head and tail entities are correct. An entity cluster is correct if the clustered mentions, the entity type and the entity identifier match a ground truth entity cluster. Finally, a mention is correct if it matches exactly a ground truth mention span.

4.2.3 Document level closed Information Extraction (DocIE)

As document-level RE does not link entities, we extend the evaluation setting to address the joint DocIE task. We introduce the **DocIE hard metric** for the E2E task: A triple is correct if the relation type and the entity clusters associated with the head and tail entities are correct. An entity cluster is correct if the clustered mentions, the entity type and the entity identifier match a ground truth entity cluster. Finally, a mention is correct if it matches exactly a ground truth mention span.

4.2.4 Inference Speed

Since we are pioneering the task of DocIE, we do not have a related work to compare REXEL’s performance in this setting. Thus, we compare REXEL’s inference speed with JEREX (Eberts and Ulges, 2021) and DWIE (Zaporojets et al., 2021) in the RE setting. We use the code released by the authors to report the inference time. Both of these works support inference only for their respective datasets, i.e., DocRED-E2E and DWIE respectively.

5 Results

We summarize all results from single runs in Table 1. Note that DWIE and KBIE (Verlinden et al., 2021) report performance on NER instead of MD and ET separately. Therefore, they are only comparable for Coref and RC in the subtask setting. In E2E RE and E2E DocIE settings, we also report REXEL’s performance on NER, which requires both the mention span and the entity type to be correct. We follow (Zaporojets et al., 2021) for the scoring mechanism for evaluating NER performance. We demonstrate that the performance of REXEL on joint tasks (RE and DocIE) is on par with task-specific learning, while being more efficient due to shared parameters and training steps.

5.1 Subtask

In order to assess the performance of each component of REXEL, we train and evaluate each subtask individually on DWIE and DocRED-E2E split. When trained on individual subtasks only, REXEL improves upon the SoTA model on DWIE by an
average of 4 F1 points while surpassing the SoTA on DocRED-E2E on all subtasks except MD. Note that JEREX and DWIE are not only the SoTA in the RE setting but also in the subtask setting.

5.2 Relation Extraction (RE)
In the E2E RE setting, we compare with three other related works: JEREX (Eberts and Ulges, 2021), DWIE (Zaporojets et al., 2021) and KBIE (Verlinden et al., 2021). JEREX and DWIE report performance on DocRED-E2E and the DWIE dataset for RE, as well as performance on each subtask, thus being directly comparable with our setting. On the other hand, KBIE only reports performance when trained for the E2E task. We do not compare with REBEL (Huguet Cabot and Navigli, 2021) since their E2E evaluation is less strict and thus is not a fair comparison to JEREX and REXEL.

We find that REXEL outperforms the baselines on DWIE for all the individual subtasks and improves upon the SoTA on the E2E RE task by almost 14 F1 points. However, on DocRED-E2E even though REXEL improves upon JEREX for the subtasks by an average of >6 F1 points the improvement does not translate into a corresponding boost in E2E RE task. This can be attributed to the false negatives prevalent in the dataset (64.6%), which penalize the model due to missing annotations (Tan et al., 2022), significantly hampering the E2E hard metric. Also, while subtask training setting involves a single task-specific loss, the E2E RE setting involves multiple losses (cf. equation (1)), which dilutes the training effort over all the sub-tasks. This explains the slight drop in the subtasks' performance when comparing models trained in the E2E RE setting against models trained in the subtask setting. However, the E2E approach yields better E2E performance than the pipeline approach as it does not suffer from the propagation of errors.

5.3 Document level closed Information Extraction (DocIE)
For both datasets, we observe that REXEL is able to scale from the E2E RE to E2E DocIE by incorporating ED. For all the subtasks we observe comparable performance between models trained for RE and DocIE, indicating that adding ED to the joint task setting does not deteriorate REXEL’s performance on individual subtasks.

On the other hand, we observe a significant drop in the E2E task because of the additional criterion in the proposed hard metric for DocIE. In this setting, a cluster is considered incorrect if its corresponding entity identifier is incorrect, thus all the triples extracted for such a cluster are considered incorrect.

5.4 Inference Speed
We report the comparison of inference speed across datasets in Table 2. REXEL is on average almost 11 times faster than the baselines (19x on DocRED and 3x on DWIE) in the E2E RE setting, i.e., without performing ED. This can be explained by our coarse-to-fine approach, which reduces training/inference time while still preserving competitive accuracy. Even in the E2E DocIE setting, REXEL remains faster than the baselines while

Table 1: Model evaluations under various training setups evaluated individually on each subtask and the end-to-end (E2E) task. N/A denotes that the model does not support evaluation for that task. The best performing models are marked in **bold** and the second best are underlined. For DocIE training, we report the first numbers for the two datasets. * DocRED end to end split augmented with ReFinED (Ayoola et al., 2022b) entity links.
performing the additional task of ED.

<table>
<thead>
<tr>
<th></th>
<th>DocRED</th>
<th>DWIE</th>
</tr>
</thead>
<tbody>
<tr>
<td>JEREX</td>
<td>344</td>
<td>N/A</td>
</tr>
<tr>
<td>DWIE</td>
<td>N/A</td>
<td>82</td>
</tr>
<tr>
<td>REXEL (RE)</td>
<td>18</td>
<td>27</td>
</tr>
<tr>
<td>REXEL (DocIE)</td>
<td>90</td>
<td>74</td>
</tr>
</tbody>
</table>

Table 2: Inference speed comparison in seconds. The best values are in **bold** and the second best are *underlined*. N/A denotes that the code release does not support inference on the target dataset.

6 Conclusion

In this work we introduce REXEL, a highly efficient and accurate end-to-end model for document-level closed information extraction. REXEL extracts facts from unstructured text which are fully linked to a reference KG for an entire document in a single forward pass. We further demonstrate that REXEL is 11 times more computationally efficient than baselines in the same setting, while improving upon the existing baselines on E2E RE by an average of 6 F1 points across datasets and across different task settings. Specifically, we improve upon the state-of-the-art on DWIE for E2E RE by almost 14 F1 points. We report the first numbers for DocIE on DWIE and DocRED-E2E augmented with entity links. We also release the latter dataset to facilitate benchmarking of future works on DocIE. Thus, the combination of accuracy, speed and scale makes REXEL suitable for being deployed to extract fully linked facts from web-scale unstructured data with state-of-the-art accuracy and an order of magnitude lower cost than existing approaches.

Limitations

One limitation of our work is that REXEL currently supports fact extraction for entities only and will miss the facts for relations where either the subject or object is a string literal. We leave the extension of REXEL to extract string literal-based facts for future work. Another limitation is that, for a given document, the context length of REXEL is limited to the maximum number of tokens that can be encoded by the base transformer, which is RoBERTa (Liu et al., 2019) in our case (see Section 3.2). This implies that the model cannot capture triples that involve very long-range dependencies that go beyond the maximal context length. In practice, we find that this problem is negligible in our case as only a few triples fall into that category for both DocRED and DWIE. However, this might have a stronger impact for other applications. In addition, this limitation is not specific to the REXEL architecture per se but is inherent to the transformer used. Finally, while the proposed DocIE hard metric provides a common ground for future benchmarks on DocIE, it may not fully align with some industrial applications where missing a few mentions within entity clusters is not critical. In such contexts, the hard metric would provide a lower bound on the performance, and other metrics can be considered for better alignment with specific application requirements.

References


Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement Learning for Relation Classification From Noisy Data. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1).


Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022. Revisiting docred – addressing the false negative problem in relation extraction. In Proceedings of EMNLP.


A Coref clustering

We detail the different approaches used for coreference clustering in the following sections.

A.1 Entity Linking

We use entity disambiguation for predicting an identifier for each mention, and then cluster mentions which have the same identifier. This approach relies on external knowledge. Also, this approach necessitates performing entity disambiguation to obtain the identifiers, which may not always be part of the task of interest, e.g., RE does not require ED.

A.2 Greedy approach

Let’s consider a set of mentions to cluster \((m_i)_{1 \leq i \leq N}\). The greedy approach comprises two stages: first, forming a similarity matrix \(S \in \mathbb{R}^{N \times N}\) from the pairwise scores, and second, forming the cluster \((C_i)_{i}\). The model is trained on the pairwise scores only. The clusters are then defined as follows:

\[
C_i := \{ m_j : \forall j \in [1, N] \text{ such that } S_{i,j} > t \text{ and } m_j \notin C_k \text{ for } 1 \leq k \leq i - 1 \}
\] (2)

where \(t \in [0, 1]\) is the coreference threshold and \(S_{i,i} = 1 \forall i \in [1, N]\). This approach iteratively considers each mention \(m_i\) and constructs a cluster based on the coreference scores between \(m_i\) and all other valid mentions, where a valid mention is one that has not yet been assigned to a cluster. Notably, each mention span is allocated to only one cluster. However, it’s crucial to acknowledge that the hard-metric constraint implies that any absent mention within a cluster renders the entire cluster invalid.

Hence, we explore an alternative approach that relaxes the constraint of a mention belonging to only one cluster. This variant, termed the Greedy approach (multiple-clusters), allows mentions to be assigned to multiple clusters simultaneously. Each cluster is then defined as follows:

\[
C_i := \{ m_j : \forall j \in [1, N] \text{ such that } S_{i,j} > t \}
\] (3)

A.3 Agglomerative Clustering

The agglomerative clustering approach also relies on forming a similarity matrix, see Figure 2. The model is trained to predict pairwise coreference scores rather than directly predicting the clusters. Put simply, the coreference resolution component of our model is optimized for predicting a similarity matrix. Then, the second stage exploits that matrix to form the clusters. The distance threshold was chosen experimentally and we did not perform hyperparameter tuning to optimize it. The coreference performance may be further improved by including the threshold in the training.

B Training Details

REXEL uses Hugging Face implementation of RoBERTa (Wolf et al., 2019) and the model is optimised using Adam (Kingma and Ba, 2015) with a linear learning rate schedule. Our main hyperparameters are represented in Table 4. Due to the high computational cost of training the model, we did not conduct an extensive hyperparameter search. Training across datasets took approximately 24 hours on average on a single machine with 1 V100 GPU. REXEL has approximately 284M parameters in its architecture setup.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning rate</td>
<td>5e-5</td>
</tr>
<tr>
<td>batch size</td>
<td>2</td>
</tr>
<tr>
<td>max sequence length</td>
<td>510</td>
</tr>
<tr>
<td>dropout</td>
<td>0.1</td>
</tr>
<tr>
<td>RC threshold</td>
<td>0.2</td>
</tr>
<tr>
<td>description embeddings dim.</td>
<td>300</td>
</tr>
<tr>
<td># training epochs</td>
<td>150</td>
</tr>
<tr>
<td># candidates</td>
<td>30</td>
</tr>
<tr>
<td># wikidata entity types</td>
<td>1400</td>
</tr>
<tr>
<td>mention transformer init.</td>
<td>roberta-base</td>
</tr>
<tr>
<td># mention encoder layers</td>
<td>12</td>
</tr>
<tr>
<td>description transformer init.</td>
<td>roberta-base</td>
</tr>
<tr>
<td># description encoder layers</td>
<td>2</td>
</tr>
<tr>
<td># RC encoder layers</td>
<td>4</td>
</tr>
<tr>
<td>RC coarse-to-fine k</td>
<td>2000</td>
</tr>
<tr>
<td># description tokens</td>
<td>32</td>
</tr>
<tr>
<td>(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)</td>
<td>((0.1, 0.005, 0.1, 0.02, 0.775))</td>
</tr>
</tbody>
</table>

Table 4: Our model hyperparameters
C Datasets

C.1 DocRED and DWIE

The DocRED dataset was constructed from Wikipedia documents, whereas DWIE was constructed from news articles. DocRED and DWIE both comprise document-level and sentence-level facts, and they are both annotated at entity-level, i.e., facts are reported between entity clusters made of several mentions, which motivates the additional coreference resolution step for extracting relations. Also, they both require different types of reasoning to extract triples living across multiple sentences (e.g., pattern recognition, logical or common-sense reasoning). We report some statistics on these dataset in Table 5. Another similarity is that both datasets have a class-imbalance problem, which increases the complexity of the RC task. More precisely, 10 relations account for about 60% of the facts in DocRED, while the 10 most frequent relations account for more than 75% of the facts for DWIE. In addition, DocRED-E2E contains some duplicate annotations, which we remove at evaluation stage following the convention introduced by (Eberts and Ulges, 2021). Likewise, DWIE contains some spurious empty clusters (see Table 6), which we remove with their associated triples following the setting adopted by (Xu and Choi, 2022).

C.2 DocRED-IE

To facilitate future works on DocIE, we release DocRED-IE, an extension of the DocRED (Yao et al., 2019) dataset further equipped with entity links, making it the second dataset to support

<table>
<thead>
<tr>
<th># Documents</th>
<th>DocRED</th>
<th>DocRED-E2E</th>
<th>DWIE</th>
</tr>
</thead>
<tbody>
<tr>
<td># Entities/doc</td>
<td>19.5</td>
<td>19.4</td>
<td>28.3</td>
</tr>
<tr>
<td># Facts/doc</td>
<td>13.2</td>
<td>12.5</td>
<td>27</td>
</tr>
<tr>
<td># Entity types</td>
<td>6</td>
<td>6</td>
<td>311</td>
</tr>
<tr>
<td># Relations</td>
<td>96</td>
<td>96</td>
<td>65</td>
</tr>
</tbody>
</table>

Table 5: Some statistics for DocRED, DocRED-E2E and DWIE, # Entities/doc and # Facts/doc refer respectively to the averaged number of entities and facts per document.

<table>
<thead>
<tr>
<th># Mentions/Entity</th>
<th>DocRED-E2E (%)</th>
<th>DWIE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>5.3</td>
</tr>
<tr>
<td>1</td>
<td>81.7</td>
<td>62.9</td>
</tr>
<tr>
<td>2</td>
<td>11.1</td>
<td>14.4</td>
</tr>
<tr>
<td>3</td>
<td>3.6</td>
<td>6.1</td>
</tr>
<tr>
<td>&gt; 4</td>
<td>3.6</td>
<td>11.3</td>
</tr>
</tbody>
</table>

Table 6: Proportion of mentions per entity cluster in DocRED-E2E and DWIE.
DocIE evaluation, thereby facilitating future research on document-level closed information extraction. DocRED-IE allows for training and evaluation in a multitask setting encompassing mention detection, entity typing, coreference resolution, document-level relation classification, and entity linking, along with any combination thereof in a joint setting, such as the end-to-end RE task and DocIE.

DocRED-IE builds on the end-to-end DocRED release introduced in (Eberts and Ulges, 2021) (DocRED-E2E). We employ a state-of-the-art entity linking model (Ayoola et al., 2022c) to populate each mention in DocRED-E2E. Statistics of the DocRED-IE dataset are shown in Table 7.

### C.3 Dataset Licenses

The DWIE dataset (Zaporojets et al., 2021) and the code has been released under GNU GPLv3 license ². Both the DocRED-E2E ³ dataset (Eberts and Ulges, 2021) and DocRED-IE are released under MIT licence.

---

²https://github.com/klimzaporojets/DWIE/blob/master/LICENSE

³https://github.com/lavis-nlp/jerex/blob/main/LICENSE